611-TD-571-001

EOSDIS Core System Project

M&O Procedures: Section 4 — Database and Replication Server Administration

Interim Update

March 2000

Raytheon Systems Company
Upper Marlboro, Maryland

Preface

This document is an interim update to the Mission Operations Procedures Manual for the ECS Project, document number 611-CD-500-001. This document has not been submitted to NASA for approval, and should be considered unofficial.

The document has been updated to include information relevant to ECS Release 5B.

Any questions should be addressed to Ken Simmons or Rodney Creecy:

Data Management Office
The ECS Project Office
Raytheon Systems Company
1616 McCormick Drive
Upper Marlboro, Maryland 20774-5301

This page intentionally left blank.

4. Database Administration

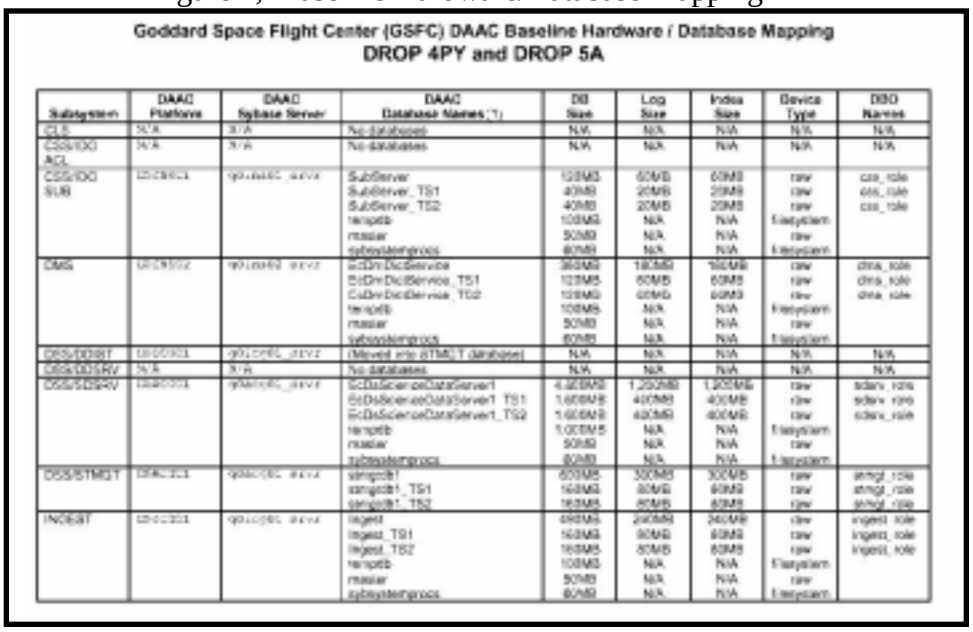
4.1 Overview of Database Administration

The Database Administrator or DBA, is the individual or group responsible for the installation, configuration, update, maintenance, and overall integrity, performance and reliability of the SQL Server database. In general, the DBA is concerned with the availability of the server, the definition and management of resources allocated to the server, the definition and management of databases and objects resident on the server, and the relationship between the server and the operating system.

4.1.1 ECS Database Environment

The database environment at ECS spans multiple databases serving numerous subsystems across several hosts. Figure 1 shows the system Baseline Hardware/Database Map for the Goddard Space Flight Center (GSFC) Distributed Active Archive Center (DAAC). There are similar mapping diagrams for each of the other DAACs. The diagram, using ECS naming conventions, lists the Subsystem name (i.e., SUB – Subscription Server; SDSRV – Science Data Server; INGEST), the Host Platform (i.e., G0INS01, G0ACG01, G0ICG01), the Sybase Server designation (i.e., g0ins01_srvr, g0acg01_srver, g0icg01_srvr), the Database Names (i.e., SubServer, EcDsScienceDataServer1, Ingest), the various database component sizes (e.g., DB size, Log size, Index size), the Device Type (raw or filesystem) and the Database Owner Names (i.e., css_role, sdsrv_role)

Figure 1, Baseline Hardware/Database Mapping



Goddard Space Flight Center (GSFC) DAAC Baseline Hardware / Database Mapping
DROP 4PY and DROP 5A

Subsystem	DAAC Platform	DAAC Sybase Server	DAAC Database Names (*)	DB Size	Log Size	Index Size	Device Type	DBO Names
CLS	N/A	N/A	No database	N/A	N/A	N/A	N/A	N/A
CSS/IDG ACL	N/A	N/A	No database	N/A	N/A	N/A	N/A	N/A
CSS/IDG SUB	[illegible]	[illegible]	SubServer	120MB	60MB	60MB	raw	css_role
			SubServer_TS1	40MB	20MB	20MB	raw	css_role
			SubServer_TS2	40MB	20MB	20MB	raw	css_role
			tempdb	100MB	N/A	N/A	filesystem	
			master	50MB	N/A	N/A	raw	
			sybsystemprocs	60MB	N/A	N/A	filesystem	
DMS	[illegible]	[illegible]	EcDmDictService	360MB	180MB	180MB	raw	dms_role
			EcDmDictService_TS1	120MB	60MB	60MB	raw	dms_role
			EcDmDictService_TS2	120MB	60MB	60MB	raw	dms_role
			tempdb	100MB	N/A	N/A	filesystem	
			master	50MB	N/A	N/A	raw	
			sybsystemprocs	60MB	N/A	N/A	filesystem	
DSS/DDIST	[illegible]	[illegible]	(Moved into ASTM_1T database)	N/A	N/A	N/A	N/A	N/A
DSS/DDSRV	N/A	N/A	No database	N/A	N/A	N/A	N/A	N/A
DSS/SDSRV	[illegible]	[illegible]	EcDsScienceDataServer1	4,800MB	1,200MB	1,200MB	raw	sdsrv_role
			EcDsScienceDataServer1_TS1	1,600MB	400MB	400MB	raw	sdsrv_role
			EcDsScienceDataServer1_TS2	1,600MB	400MB	400MB	raw	sdsrv_role
			tempdb	1,000MB	N/A	N/A	filesystem	
			master	50MB	N/A	N/A	raw	
			sybsystemprocs	60MB	N/A	N/A	filesystem	
DSS/STMGT	[illegible]	[illegible]	stmgmtdb1	600MB	300MB	300MB	raw	stmgt_role
			stmgmtdb1_TS1	160MB	80MB	80MB	raw	stmgt_role
			stmgmtdb1_TS2	160MB	80MB	80MB	raw	stmgt_role
INGEST	[illegible]	[illegible]	ingest	680MB	240MB	240MB	raw	ingest_role
			ingest_TS1	160MB	80MB	80MB	raw	ingest_role
			ingest_TS2	160MB	80MB	80MB	raw	ingest_role
			tempdb	100MB	N/A	N/A	Filesystem	
			master	50MB	N/A	N/A	raw	
			sybsystemprocs	60MB	N/A	N/A	filesystem	

The Hardware/Database Mapping document, 920-TDx-009-Rev*xx*, along with many other key reference documents can be accessed through the Pete.hitc.com website:

http://cmdm.east.hitc.com. Select the "ECS Baseline" link button, and then the "Technical Documents" button to view this wealth of system data.

In addition to the fundamental database design, ECS operates on a concept of mutually exclusive, functionally identical modes. The main mode that interacts with live data and customers is called the Operational mode (OPS). Other modes available at the DAACs are nominally called TS2, TS1, and SHARED. The SHARED mode contains files common to all modes. The TS2 and TS1 modes are used to implement and test new functionality for both COTS and CUSTOM code. After modifications are installed and successfully tested in a non-OPS mode, they are promoted to the next mode level and ultimately upgraded into the OPS mode. This concept enables uninterrupted operation for live data and user interaction while simultaneously field-testing new code. Figure 2 shows this multi-mode directory structure. The OPS mode is shown here and is identical for the TS2 and TS1 modes.

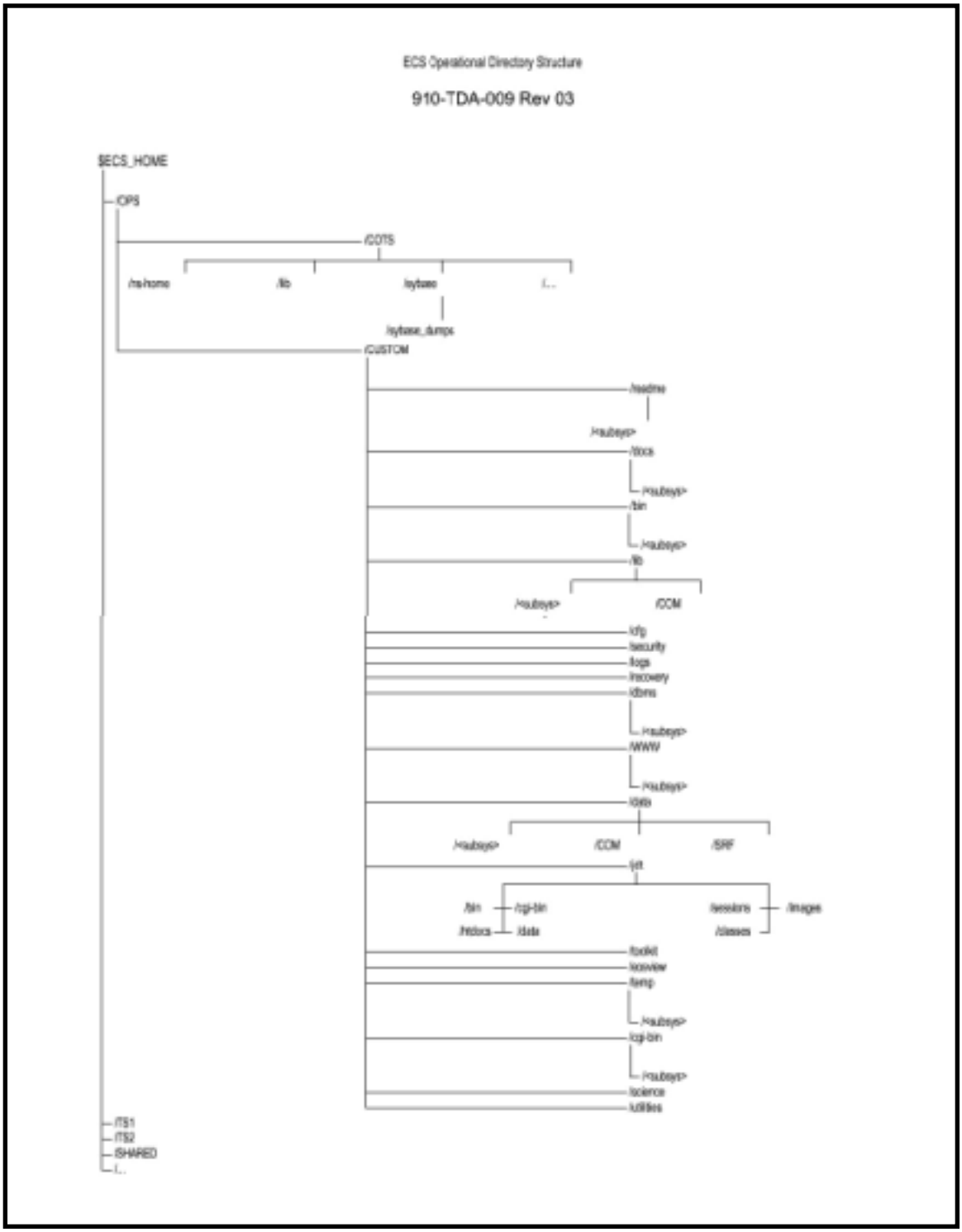


Figure 2, ECS Operational Directory Structure

4.1.2 Sybase Adaptive Server Enterprise

The version of Sybase Adaptive Server Enterprise installed in drop 5B on the HP, SUN and SGI platforms is ASE 11.5.1. Version numbers for this COTS software and all other software products are published in the COTS SOFTWARE VERSION BASELINE REPORT, 910-TDA≠ 003-Rev*xx*. A more complete listing of software and the individual host installation content is provided in the DAAC-specific Hardware/Software Map, document 920-TDx-002-Rev*xx*. As upgrades are released and installed, version status will be reflected in these documents.

4.1.3 Database Schemas

All database designs in ECS are thoroughly documented in the 311 Series of documents, <SUBSYSTEM> Database Design and Schema Specifications for the ECS Project. These individual subsystem documents provide the DBA with a complete description of each database including:

Physical Data Model Entity Relationship Diagram

Tables

Columns

Column Domains

Rules

Defaults

Views

Integrity Constraints

Triggers

Stored Procedures

The Schema documents also provide Performance and Tuning Factors, Database Security information, Scripts, and Entity Relationship Diagram Keys. Figure 3 is a portion of the Entity Relationship Diagram for the INGEST subsystem. To access this document, and the other subsystem 311 Series documents, use the following URL: http://edhs1.gsfc.nasa.gov/. From the ECS Data Handling Homepage select the "Document Catalog" link and then the "Design Documents and Specifications" link. From this point you can select the relevant subsystem document.

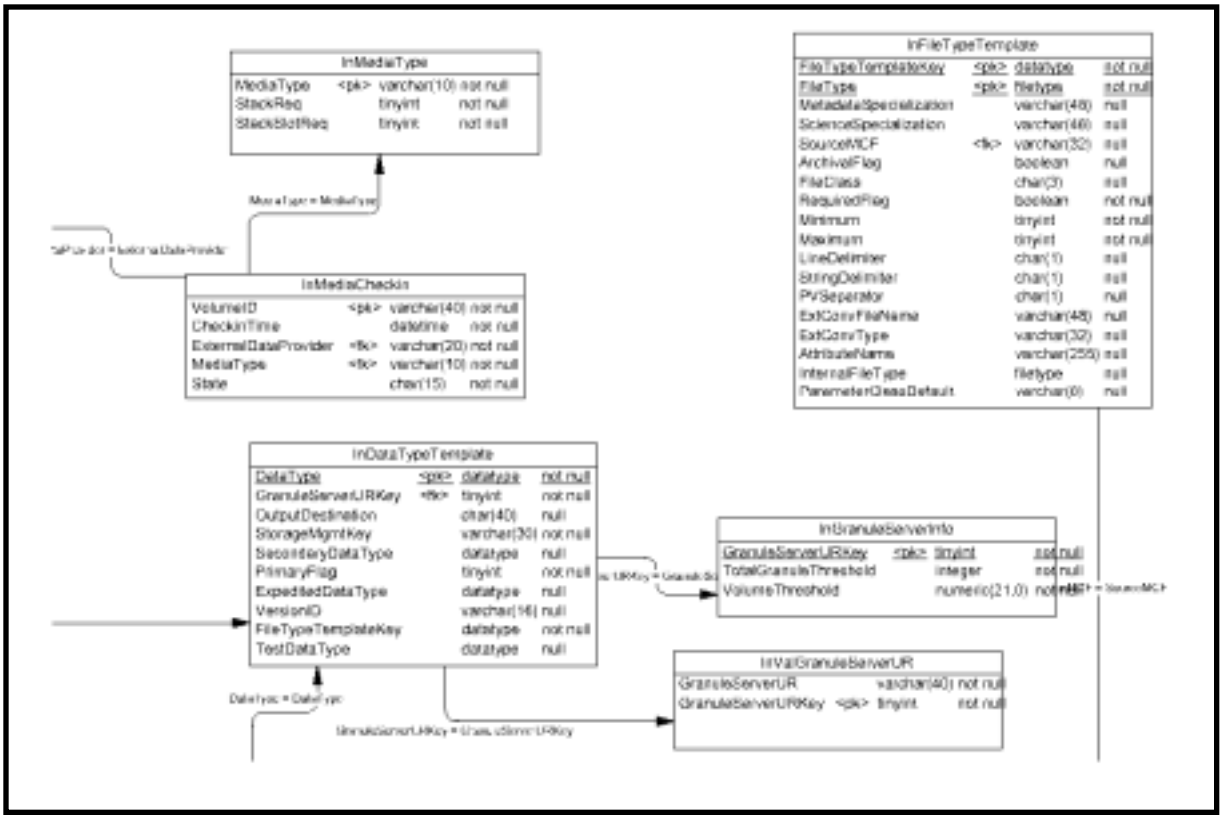


Figure 3, Partial Entity Relationship Diagram - INGEST

DAAC Database Configurations

The key factors that determine optimum performance for any database are its configuration parameters. A new document available on the Pete Server (http://cmdm.east.hitc.com), SYBASE SQL Server 11.0.x, ALL DAAC Database Configurations (910-TDA-021-Rev00), provides DBAs with detailed data and recommendations for Sybase configurable parameters. In addition to providing default values and the DAAC-specific parameter values for each host, it also describes the Sybase Segment naming conventions to be used at each site for assigning Sybase disk devices to databases. Future versions of this document will capture and baseline the disk devices at each DAAC and the interface file listings at each DAAC.

Figure 4 is an example of DAAC-specific configuration parameters for Goddard Space Flight Center on host g0msh08.

Host Name: g0msh08

Parameter Name	Memory Used	Default Value	Configured Value	Run Value
number of devices	8	10	20	20
number of locks	469	5000	5000	5000
number of remote connections	33	20	20	20
number of remote logins	22	20	20	20
total memory	15000	7500	7500	7500
max online engines	355	1	1	1
number of user connections	2121	25	25	25
procedure cache percent	838	20	20	20
number of open databases	396	12	12	12
number of open objects	489	500	500	500
stack size	2075	34816	34816	34816

Figure 4, DAAC-Specific Configuration Parameters

Database Disk Partitioning

System documentation also provides graphic depictions of server disk partitioning for all of the hosts (e.g. Ingest, MSS, CSS, PDPS DBMS). The 922-TDx-0xx-Rev*00* series of documents (available on the PETE server) provide a block diagram of the disk partitioning for each of the servers (Figure 5) and also secondary tables (Figure 6) describing the physical break-down of the individual disks including Slice, Start Block, Total Blocks, Start MB, Total MB, XLV Name, Mount, and Type.

Ingest Server Disk Partitioning Diagram
g0icg01 & g0icg02

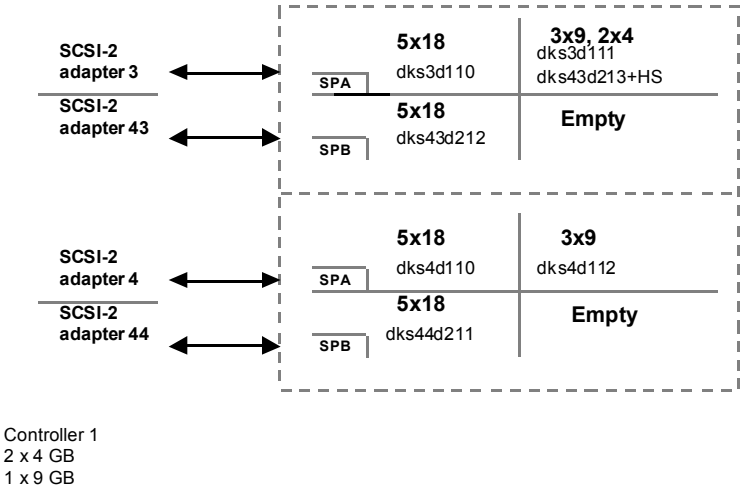


Figure 5, Server Disk Partitioning Block Diagram

dks3d111, 2x0 GB Raid Level 1, Mirrored

Slice	Start Block	Total Blocks	Start MB	Total MB	XLV Name	Mount	Type
0	2048	1024000	1	1	ingestlog		xlv
1	1026048	512000	501	501	sybsecurity		xlv
2	1538048	256000	751	125	sybmaster		xlv
3	1794048	512000	876	250	sybsecarchive		xlv
4	2306048	512000	1126	250	ddistlog		xlv
5	2814048	1024000	1376	500	stmgtlog		xlv
6	3842048	5120000	1876	2500	sybase_dumps		xlv
7	9862048	8264978	4376	4036	spare1		xlv
8	0	2018	0	1	n/a		Volhdr
10	0	17227026	0	8412	n/a		volume

Figure 6, Ingest Server Disk Partitioning Diagram

4.2 SQL Server Environment

4.2.1 Naming Conventions,

As one of the most important, yet least applied concepts, naming conventions are presented in this chapter by examples according to the following rules.

Rule1: Regardless of the length of the name, it should indicate the function and/or content of the object

Rule_2: Only easily understandable abbreviations should be used

Rule_3: Parts of names are separated by underscores "_", only one optional suffix is permitted (appended to the name by a . ".")

Rule_4: The full path of the object is considered to be part of the name

The names of the databases and tables themselves may or may not follow the above rules, these rules are specifically for the DBA to work with SQL Server objects, and files in the UNIX environment.

All **COTS** software is installed in the /usr/ecs/OPS/COTS directory.

All **SYBASE** software is located in the Sybase home directory (**$SYBASE**).

All backups are located in **$SYBASE**/sybase_dumps directory, which may or may not be on a separate physical disk.

Note

It is strongly recommended that backups be stored on a separate physical disk.

The database dumps are kept for a period of 2 days and also stored on a disk by Networker everyday. The database dumps are named as follows:

dbname.dat_YYMMDDHHMM.Z

where MMDDHHMM is the "sortable" eight digit month, day, hour, and minute. For example, on the date this chapter was written, a backup directory called **backups_for_99021100024.Z**

All SQL script files have the extension **.sql** as a suffix. Their names reference the objects they create or functions they perform, and are all located in **$SYBASE**/scripts.

SQL statement must follow precise syntactical and structural rules, and may include only SQL keywords, identifiers (names of databases, tables, or other database objects), operators, and constants. The characters that can be used for each part of a SQL statement vary from installation to installation and are determined in part by definitions in the default character set that version of the server uses.

For example, the characters allowed for the SQL language, such as SQL keywords, special characters, and Transact-SQL extensions, are more limited than the characters allowed for

identifiers. The set of characters which may be used for data is much larger and includes all the characters that can be used for the SQL language or for identifiers.

The sections that follow describe the sets of characters that can be used for each part of a statement. The section on identifiers also describes naming conventions for database objects.

4.2.1.1 SQL Data Characters

The set of SQL data characters is the larger set from which both SQL language characters and identifier characters are taken. Any character in SQL Server's character set, including both single≠ byte and multibyte characters, may be used for data values.

4.2.1.2 SQL Language Characters

SQL keywords, Transact-SQL extensions, and special characters such as the comparison operators > and <, can be represented only by 7-bit ASCII values A- Z, a -z, 0-9, and the following ASCII characters:

4.2.1.3 Identifiers

Conventions for naming database objects apply throughout SQL Server software and documentation. Identifiers can be up to 30 bytes in length, whether or not multibyte characters are used. The first character of an identifier must be declared as an alphabetic character in the character set definition in use on Server.

The @ sign or _ (underscore character) can also be used. The @ sign as the first character of an identifier indicates a local variable.

Temporary table names must either begin with # (the pound sign) if they are created outside tempdb or be preceded by "tempdb..".

Table names for temporary tables that exist outside tempdb should not exceed 13 bytes in length, including the number sign, since SQL Server gives them an internal numeric suffix.

After the first character, identifiers can include characters declared as alphabetic, numeric, or the character $, #, @, _, ¥ (yen), or £ (pound sterling). However, you cannot use two @@ symbols together at the beginning of a named object, as in "@@myobject." This naming convention is reserved for global variables, which are system-defined variables that SQL Server updates on an ongoing basis.

The case sensitivity of SQL Server is set when the server is installed and can be changed by a System Administrator. To see the setting for your server, execute this command: sp_helpsort

4.2.1.4 Delimited Identifiers

Delimited identifiers are object names enclosed in double quotes. Using delimited identifiers allows you to avoid certain restrictions on object names. You can use double quotes to delimit table, view, and column names; you cannot use them for other database objects.

Delimited identifiers can be reserved words, can begin with non-alphabetic characters, and can include characters that would not otherwise be allowed. They cannot exceed 28 bytes.

Before creating or referencing a delimited identifier, you must execute:

set quoted_identifier on

The names of database objects need not be unique in a database.

However, column names and index names must be unique within a table, and other object names must be unique for each owner within a database. Database names must be unique on SQL Server.

If you try to create a column using a name that is not unique in the table or to create another database object such as a table, a view, or a stored procedure, with a name that you have already used in the same database, SQL Server responds with an error message.

You can uniquely identify a table or column by adding other names that qualify it, that is, the database name, the owner's name, and, for a column, the table name or view name. Each of these qualifiers is separated from the next by a period:

database.owner.table_name.column_name

database.owner.view_name.column_name

The same naming syntax applies to other database objects. You can refer to any object in a similar fashion:

If the quoted_identifier option of the set command is on, you can use double quotes around individual parts of a qualified object name.

Use a separate pair of quotes for each qualifier that requires quotes.

For example, use:

database.owner."table_name"."column_name"

rather than:

database.owner."table_name.column_name"

The full naming syntax is not always allowed in create statements because you cannot create a view, procedure, rule, default, or trigger in a database other than the one you are currently in. The naming conventions are indicated in the syntax as:

[[database.]owner.]object_name or: [owner.]object_name

The default value for owner is the current user, and the default value for database is the current database. When you reference an object in SQL statements, other than create statements, without qualifying it with the database name and owner name, SQL Server first looks at all the objects you own, and then at the objects owned by the Database Owner, whose name in the database is "dbo." As long as SQL Server is given enough information to identify an object, you need not type every element of its name. Intermediate elements can be omitted and their positions indicated by periods:

database..table_name

You must include the starting element, in this case, database, particularly if you are using this syntax when creating tables. If you omit the starting element, you could, for example, create a table named ..mytable. This naming convention prevents you from performing certain actions on such a table, such as cursor updates.

When qualifying a column name and a table name in the same statement, be sure to use the same naming abbreviations for each; they are evaluated as strings and must match or an error is returned.

4.2.1.5 Identifying Remote Servers

You can execute stored procedures on a remote SQL Server, with the results from the stored procedure printed on the terminal that called the procedure. The syntax for identifying a remote server and the stored procedure is:

[execute] server.[database].[owner].procedure_name

You can omit the execute keyword when the remote procedure call is the first statement in a batch. If other SQL statements precede the remote procedure call, you must use execute or exec. You must give the server name and the stored procedure name. If you omit the database name, SQL Server looks for procedure_name in your default database. If you give the database name, you must also give the procedure owner's name, unless you own the procedure or the procedure is owned by the Database Owner.

If the server name in interfaces is in uppercase letters, you must use it in uppercase letters in the remote procedure call.

In all cases throughout this chapter, when actual examples are provided, those which reference UNIX commands will be preceded by a “%”, and those that reference SQL statements will be preceded by a number and a “>” (e.g. 1>sp_help tablename).

The terms described in the following table will be used throughout this chapter.

Table 4.2-1. SQL Server General Definitions

Term	Definition
SQL Server	The server in the Sybase client/server architecture. SQL Server manages multiple databases and multiple users, keeps track of the actual location of data on disks, maintains mapping of logical data description to physical data storage, and maintains data and procedure caches in memory.
Client	SYBASE Open Client software located in the /tools/sybOCv(TBD) directory for SUN and HP platforms SYBASE Open Client software located in the /tools/sybOCv(TBD) directory for SGI platform
Backup Server	Similar to the dataserver, it uses a separate UNIX process to off load the cycles associated with DUMP and LOAD commands
backups	The set of UNIX files containing full database dumps, transaction log dumps, and dbcc output
dbcc	Database Consistency Checker - a utility program designed to check the logical and physical consistency of a database
sybase root directory	/usr/ecs/OPS/COTS/sybase, this is the home directory for all SYBASE software and related products and is referenced both in UNIX and in the rest of this document as **$SYBASE**
interfaces file	Lists the names and access paths for all servers and backup servers. This file is located in the **$SYBASE**
sa	System Administrator login, this is the superuser of the SQL Server
scripts	UNIX script programs located in **$SYBASE**/scripts and related subdirectories {$ecs_Home}/{mode}/custom/dbms/{subsystem}
showserver	A utility invoked at the UNIX command prompt to display active servers, located in **$SYBASE**/install.
SQL scripts	SQL and command statements located in **$SYBASE**/scripts and related subdirectories and /{$ecs_Home}/{mode}/custom/dbms/{subsystem}
Server Name	The name of the database server for a specific application in different modes **EX. -** PDPS application database server in OPS mode EX.- Pdps_TS1 in TS1 mode EX. -pdps _TS2 in TS2 mode
Port Numbers	The port number to be utilized by the above listed servers.
Release Directory	$SYBASE
SQL	Structured Query Language

4.2.2 SQL Server Directory Structure

The **sybase** directory structure is described in the following table. Subdirectories under the **scripts** can contain template files with easy to modify examples of SQL and SQL command syntax.

Table 4.2.2-1. SYBASE Directory Structure

Directory	Contains
$SYBASE/bin	Utilities necessary to load, run, and access the server
$SYBASE /install	Files used to start and initialize dataservers, backupserver and to record server messages (errorlogs)
$SYBASE /lib	db-lib, ct-lib, and xa-lib client library files used by applications to gain access to the server (local to server) *__Applications use automounted libraries__.
$SYBASE /scripts	Root directory for all script files executed on the server
$SYBASE /sybase_dumps	Root directory that contains all backup subdirectories, it is recommended, but not required, that this directory be on a separate physical disk. Dumps both database and transaction logs. **Backups are stored on disk in the backup subdirectories.
backup subdirectories $SYBASE /sybase_dumps/dumps $SYBASE /sybase_dumps/trans $SYBASE /sybase_dumps/dumps/logs $SYBASE /sybase_dumps/trans/logs $SYBASE /sybase_dumps/Week1 $SYBASE /sybase_dumps/Week2 $SYBASE /sybase_dumps/Week1/logs $SYBASE /sybase_dumps/Week2/logs	A cron job is run at night to move data from the current (week1) directory to the previous (week2) directory. Then, a dump of the databases and transaction logs is executed and is stored in the current directory. All logs are written to the log directory. Files are saved using the following naming convention:: dbname.dat.YYMMDDHHMM.Z - full database dumps dbname.tran.YYMMDDHHMM.Z - full transaction log dumpsdbname_backup.log. dbname_ERR.log.MMDDHHMM - Error log filesdbname_dbcc.log. MMDDHHMM
**xxdmh02 serves as a remote Backup Server	**xx are the 2 letter codes to identify a DAAC site (i.e., g0 = Goddard)

4.2.3 SQL Server Installation

SYBASE SQL Server Version (TDB) has been installed and configured by the ECS Installation Staff. Shared memory and disk resources have been allocated and configured by the System Administrator, and both the client and server portions have been set up by the DBA prior to shipment. The following table describes parameters and options used during installation.

Table 4.2.3-1. SQL Server Parameters and Options

Parameters Name	Brief Explanation/Settings
Retry Count	5 seconds
Retry Delay	5 seconds
Master device	28 Mb raw partition
Master Device Location	
Backup Server Name	SYB_BACKUP

sybsystemprocs	$SYBASE/devices/(MachineName)_sybprocs.dat, 19 Mb and on it's own device
Errorlog	$SYBASE/install/mode.errorlog (**mode** indicates the application)
Current default language	us_english
Current default character set	iso_8859-1 (Latin-1)
Current sort order	Binary ordering, for the ISO 8859/1 or Latin-1 character set (iso_1).
Internal auditing	On
sybsecurity database size	175 Mb - Varies – depends on disk allocations
sybsecurity device	sybsecurity, positioned on a 175 Mb raw partition

The installation script files are located in the **$SYBASE**/install directory. SQL Server installation is performed by an authorized user with the **sybinit** utility also located in the **$SYBASE**/install directory. See your UNIX System Administrator and the SYBASE SQL Server Installation Guide.

4.3 Database Administrator Responsibilities

The following subsections detail the most common functions that a DBA will perform.

4.3.1 Startup of SQL Server

Use **startserver** to start an SQL Server and/or a Backup Server. This command can only be issued by the **Sybase** user.

Syntax: % **startserver** [-f runserverfile]

The "runserverfile" is contained in the **$SYBASE**/install directory.

Note

SQS server should be started after the SQL Server

4.3.2 Shutdown of SQL Server

Use **shutdown** to bring the server to a halt. This command can only be issued by the Sybase System Administrator (sa).

Syntax:1> **shutdown [backup_server_name]]** [with] [wait] [with nowait]

2> **go**

The "with wait" is the default option. This option brings SQL Server down gracefully.

The "with nowait" option shuts down the SQL Server immediately without waiting for currently executing statements to finish.

If you do not give a server name, shutdown shuts down the SQL Server you are using.

When you issue a shutdown command, SQL Server:

Disables logins, except for System Administrators

Performs a checkpoint in each database, flushing pages that have changed from memory to disk

Waits for currently executing SQL statements or procedures to finish

In this way shutdown minimizes the amount of work that automatic recovery must do when you restart SQL Server.

To see the names of the Backup Servers that are accessible from your SQL Server, execute

sp_helpserver. Use the value in the name column in the shutdown command. You can only shut down a Backup Server that is:

Listed in sysservers on your SQL Server, and

Listed in your local interfaces file.

Note 1

It recommended that "with wait" option be used. This allows executing statements to finish.

Also it is recommended that you perform a checkpoint of all database prior to shutdown.

Note 2

SQS server should be started after the SQL Server

4.3.3 Showing SQL Server(s)

Use **showserver** to determine whether the SQL Server(s) and/or Backup Server(s) are running.

Syntax: % **showserver**

The "**showserver**" is contained in the **$SYBASE**/install directory

Example: UNIX processes running the various servers:

```
  UID   PID  PPID   C    STIME  TTY      TIME COMD
sybase  671   669   80   Apr 17   ?        80:05  /usr/ecs/OPS/COTS/Sybase/bin/dataserver   -d
/dev/rdsk/c1t0d0s1 -g0sps06_srvr
sybase  665   663   80   Apr 17   ?        50:02  /usr/ecs/OPS/COTS/sybase/bin/backupserver  ≠
g0sps06_backup -e/usr/ecs/OPS
```

4.4 Allocation of Resources

SQL Server can make reasonable default decisions about many aspects of storage management, such as where databases, tables, and indexes are placed and how much space is allocated for each one. However, the System Administrator has ultimate control over the allocation of disk resources to SQL Server and the physical placement of databases, tables, and indexes on those resources.

4.4.1 Creating Logical Devices

A logical device is created when the UNIX System Administrator determines that new disk space is available for use by SYBASE software, databases, transaction logs, and/or backups. Either raw disk partitions or UNIX filesystem partitions can be used to create a logical device. The creation of a logical device is a mapping of physical space to a logical name and virtual device number (**vdevno**) contained in the SQL Server **master** database. The **disk init** command is used to initialize this space. After the disk initialization is complete, the space described by the physical address is available to SQL Server for storage, and a row is added to the **sysdevices** table in the **master** database.

A System Administrator initializes new database devices with the disk init command.

Disk Init does the following: Maps the specified physical disk device or operating system file to a database device name

Lists the new device in master..sysdevices

Prepares the device for database storage

Note

Before you run disk init, see the SQL Server installation and configuration guide for your platform for information about choosing a database device and preparing it for use with SQL Server. You may want to repartition the disks on your computer to provide maximum performance for your Sybase databases.

Disk init divides the database devices into allocation units of 256 2K pages, a total of 1/2MB. In each 256-page allocation unit, the disk init command initializes the first page as the allocation page, which will contain information about the database (if any) that resides on the allocation unit.

Note

After you run the disk init command, be sure to use dump database to dump the master database. This makes recovery easier and safer in case master is damaged. If you add a device and fail to back up master, you may be able to recover the changes with disk reinit.

Syntax: disk init

name = "device_name" ,

physname = "physicalname" ,

vdevno = virtual_device_number ,

size = number_of_blocks

[, vstart = virtual_address ,

cntrltype = controller_number]

4.4.1.1 Example of Creating a Logical Device

A raw partition on a RAID device has been made available to SQL Server by the UNIX System Administrator. Essentially, the actual name of the raw device **c2t0d1s3** has had it's ownership changed to **sybase** and it's group changed to **user**.

1 . In **$SYBASE**/scripts/create.devices, DBA makes a script file from the template.

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.devices

% cp template.sql data_dev1.sql

2 . Appropriate items are modified so that the script file resembles the following:

1> disk init

2> name = "data_dev1",

3> physname = "/dev/rdsk/c2t0d1s3",

4> vdevno = 3,

5> size = 128000

6> go

7> sp_helpdevice data_dev1

8> go

3 . DBA runs the script from the UNIX command prompt:

Syntax: % isql -Usa -S**servername** -idata_dev1.sql -odata_dev1.out

4 . DBA checks the data_dev1.out file for success

4.4.2 Creating and Altering Databases

A user database is created by the DBA with a script containing the **create database** command. A database is created on one or more physical devices. Specifying the device is optional - but highly recommended. When indicating the device, you use the logical name you specified as part of a **disk init** (described above). Unlike the **disk init** command, the size of the database data and log components is specified in MB instead of 2K pages.

4.4.2.1 Example of Creating a Database

The logical device **data_dev1** has been created (as above) along with another device called **tx_log1** (for transaction logging).

1. In **$SYBASE**/scripts/create.databases directory, DBA makes a script file from the template.

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.databases

% cp template.sql userdb.sql

2. Appropriate items are modified so that the script file resembles the following:

1> create database UserDB on data_dev1 = 100 log on tx_log1 = 50 [with override]

2> go

3> sp_helpdb UserDB

4> go

3 DBA runs the script from the UNIX command prompt:

Syntax: %isql -Usa -S**servername** -iuserdb.sql -ouserdb.out

4 DBA checks the userdb.out file for success

4.4.2.2 Example of Altering a Database

The user database **UserDB** has run out of space and it has been determined that it should be increased by 50MB.

1 In **$SYBASE**/scripts/create.databases, DBA creates a script file containing the ALTER DATABASE command (named alter_userdb.sql)

Syntax: Alter database UserDB on data_dev3 = 50

2 DBA runs the script from the UNIX command prompt:

Syntax: % isql -Usa -S**servername** -ialter_userdb.sql -oalter_userdb.out

3 DBA checks the alter_userdb.out file for success

4.4.2.3 Data Placement - Segmentation

Segments are named subsets of the database devices available to a particular SQL Server database. Segment names are used in **create table** and **create index** commands to place tables or indexes on specific database devices. Using segments allows the DBA to better control the size of database objects and may improve performance by spreading i/o more evenly across devices.

Once the database device exists and is available, the segment can be defined with the system stored procedure **sp_addsegment**.

Syntax: sp_addsegment segname, dbname, devname

After the segment has been defined in the current database, the **create table** or **create index** commands use the optional clause “on segment_name” to place the object on a particular segment.

Syntax: create table table_name (column_name datatype ...) [on segment_name]

create [clustered | nonclustered] index index_name on table_name (columns)

Use **sp_helpdb** database_name to display the segments defined for that database.

Use **sp_helpsegment** segment_name to list the objects on the segment and show the mapped devices.

4.4.2.3.1 Example of Creating a Segment

The DBA receives a request to create a segment for the storage of the DATA_INFO table indexes in the pdps_db_ops database, on a separate physical disk. Two devices **data_dev1** and **data_dev2** have already been created and are located on different physical disks.

1 . In **$SYBASE**/scripts/create.segments directory, DBA makes a script file from the template.

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.segments

% cp template.sql segments_dev1.sql

2 . The script file is modified so that it resembles the following:

1> sp_addsegment seg1_dev1, pdps, data_dev1

2> sp_addsegment seg1_dev2, pdps, data_dev2

3> go

3 . DBA runs the script from the UNIX command prompt:

Syntax: ∞ %isql -Usa -S**servername** -ipdps_db_ops_segments.sql \

-opdps_db_ops_segments.out

4 . DBA checks the pdps_segments.out file for success

5 . When the table and indexes are created according to the instructions in section 4.4.6, the “on seg1_dev1” must be appended to the DATA_INFO.sql **create table** statement, and the “on seg1_dev2” must be appended to the DATA_INFO_indexes.sql CREATE INDEX statement.

Syntax: **create index** DATA_INFO_IDX on DATA_INFO (DI_ID) on SEG1_DEV2

4.5 Loading a database you have created into a different database:

Occasionally, you may want to create an exact copy of a database of you system. First, dump the existing database. Then create a database to load with this dump. The database does not have to be the same size as the original. The only requirement is that the destination database must be at least as large as the dumped database and have the same beginning fragments as the original database. This information can be obtained from saved database creation scripts, or by running the following command:

select segmap,'Size in MB'=size/512 from sysusages where dbid= db_id("database_name")

Example:

suppose your database was created with the following statement:

create database dbname on datadevice1 = 1000,

log on Logdevice1 = 200

go

alter device dbname on datadevice2 = 500 running:

select segmap,'Size in MB'=size/512 from sysusages

where dbid= db_id("dbname")

would return:segmap Size in MB

3 1000

4 200

3 500

You could create a 3GB database as follows and load your database into it (using "for load" option will shorten database load time):

create database newdatabase on datadevice3 = 1000 log on logdevice3 = 200

for load

go

alter database newdatabase on datadevice 3=500 for load go

alter database newdatabase on datadevice4=300 for load go

alter database newdatabase on datadevice5=1000 for load go

load database newdatabase from dbname_dump go

4.6 Monitoring Space Usage

4. 6.1 Thresholds

Thresholds are defined on segments to provide a free space value at which a procedure is executed to provide a warning or to take remedial action.

Use **sp_addthreshold** to define your own thresholds:

sp_addthreshold database_name, segment_name, free_space, procedure_name

where free_space is the number of free pages at which the threshold procedure executes; procedure_name is the stored procedure which the threshold manager executes when the number of free pages falls below the free_space value. Please see the section on Auditing later in this chapter for an example of Thresholds.

Example of Threshold Commands mentioned above:

Sp_addthreshold CustomerDB, "default", 10230, CustDefaultSegWarn

4.7 Creating Database Objects

For special cases, creation (and modification) scripts are stored in **$SYBASE**/scripts/scriptname. There should be a template for each type of object to be created.

4.7.1 Example of Creating a User Table

The DBA has received a request to create a new table in the pdps_db_ops database called **PGE_Statistics** which has three column, pge_id, pge_statistic_type, and pge_statistic.

1 . In the **$SYBASE**/scripts/create.db_objects directory, DBA creates a script file from the proper template.

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.db_objects

% cp table_template.sql PGE_Statistics_table.sql

2 . Appropriate items are modified so that the script file resembles the following:

```
1> create table PGE_Statistics (
2> pge_id                    int,
3> pge_statistic_type   int,
4> pge_statistic             float )
        5> go
        6> sp_help PGE_Statistic
        7> go
```

3 . DBA runs the script from the UNIX command prompt:

Syntax: %isql -Usa -S**servername** -iPGE_Statistics_table.sql \

-oPGE_Statistics_table.out

4 . DBA checks the PGE_Statistics_table.out file for success

Other objects are created in like manner but are not included here due to space considerations.

4.8 Creating and Managing Logins and Roles

Earlier versions of SQL Server administrative responsibilities needed to be executed by and individual logged in –literally- as sa. Now specific user logins can be assigned components of administrative responsibility, enabling you to track and audit administrative activities.

The three roles are sa_role (systems administrator) for administrative tasks, sso_role(site security officer) for security tasks, and oper_rol (operator) for backup and recovery tasks.

In order to connect to a SQL Server a login must be created by the System Administrator or a system security officer. Login details are stored in the syslogins table in the **master** database.

The system stored procedure **sp_addlogin** adds new login names to the server but does not grant access to any user database.

Syntax: **sp_addlogin** login_name, password, [,default database ,language, fullname]

In order to gain access to a database, the System Administrator, system security officer, of the specific database owner must "add" the user with the **sp_adduser** system stored procedure.

Syntax: 1> **sp_adduser**login_name [username, group_name]
2> go

4.8.1 Example of Creating a Login and Granting Database Access

The DBA has received a request to authorize John Q. Public to the pdps_db_ops database.

***It is a good practice to have a default_db, when you create a user account.**

1 . In the **$SYBASE**/scripts/create.users directory, DBA creates a script file containing the sp_addlogin command (named public.sql)

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.users

% cp template.sql public.sql

2 . DBA modifies appropriate fields so that the script resembles the following:

1> sp_addlogin jpublic,jpublic, default_db

2> go

3> use pdps (OPS mode) 4> go

5> sp_adduser jpublic

6> go

7> sp_helpuser

8> go

3 . DBA runs the script from the UNIX command prompt:

Syntax: % isql -Usa -S**servername** - public.sql -opublic.out

4 . DBA checks the public.out file for success

4.9 Permissions

Permissions are used to control access within a database. The DBA uses the **grant** and **revoke** statements to accomplish this. There are two types of permissions within a database, **Object** and **Command**. In general, **Object** privileges control select, insert, update, delete, and execute permissions on tables, views, and stored procedures. **Command** permissions control access to the **create** statements for databases, defaults, procedures, rules, tables, and views.

The syntax for the **grant** and **revoke** statements are quite similar:

grant {all [privileges] | command_list }

to { public | name_list | role_name }

revoke {all [privileges] | command_list }

from { public | name_list | role_name }

4.9.1 Example of Granting Privileges to a Specific User

The DBA receives a request that John Q. Public should be able to read the DATA_INFO table and read and update the SUBSCRIPTION_NOTIFICATION TABLE.

Syntax: 1> **grant** select on DATA_INFO to jpublic

2> **grant** select, update on SUBSCRIPTION_NOTIFICATION to jpublic

go

Note: It is recommended that the DBA store these command in a ".sql" file in the **$SYBASE**/scripts/create.db_objects directory, along with their results.

4.10 Backup and Recovery

Table 4.10-1. Backup and Recovery Definitions

Term	Definition
Backup Script Components	Located in the **$SYBASE** directory, they include: sybasedump, dmpdb_trns, copy_daily_dumps_to_week1, copy_daily_dumps_to_week2
Backup files	Defined in Table 4.2-2, the location of these files has been determined during server setup
Backup Statements	Generated from the sql in sybasedump these include calls to dbcc, Dump Database, and Dump Transaction commands
Backup Subdirectory	The only directory level underneath of the Backup Directory, defined in Table 4.2-2.
Backup Summary	An extraction of the successful Dump messages along with any errors generated by the Backup Statements stored in the Backup Subdirectory.

4.10.1 Automatic Backups

The following are the list of all procedures and scripts files that are currently being used for Sybase backups. There are cron jobs running at all sybase **servers** that have SQL server installed. All dump files are currently written to LOCAL machine. The site DBA is responsible for configuring the backup dump to the REMOTE sybase directory.

To check if the crontab is up and running, enter:

> crontab -l

Example of the output:

019 * * 1-6 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_DumpDb

012 * * 1-6 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_DumpTran

021 * * 1-5 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_CkErrorLog

NOTE:

If the crontab is not running enter:

> crontab /usr/ecs/OPS/COTS/sybase/run_sybcron

The following files will be installed by EcCoAssist to the /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin directory:

EcCoDbSyb_README

EcCoDbSyb_DumpDb

EcCoDbSyb_DrumpTran

EcCoDbSyb_DbStat

EcCoDbSyb_SedFile

EcCoDbSyb_DboMail

EcCoDbSyb_SetupKsh

EcCoDbSyb_CkErrorLog

EcCoDbSyb_tran_log.awk

SCRIPTS	**DESCRIPTIONS**
EcCoDbSyb_SetupKsh∞	This file contains the SYBASE and DSQUERY (server) environment setup. This file is call by EcCoDbSyb_DumpDb, EcCoDbSyb_DrumpTran, and EcCoDbSyb_CkErrorLog scripts.
EcCoDbSyb_DumpDb∞	This script contains the code to dump the databases. First, it checks for any DBCC error on the master database, if there is any error on the master, the script sends an email to the DBA and exit the program. If the master database dump was successfull, then the rest of the databases are dumped. Each database has a DBCC check, if there is any error on the database then the database is NOT dumped and an email is send to the DBA. At the end, an status email is send, providing all the database names that were succefully dumped
EcCoDbSyb_DumpTran∞	This script contains the code to dump the transaction logs. This dumps the transaction logs for each database, it check the error log file, if the error Msg is 4207 or 4221 it will do a dump of the database firt, then it will do the trasaction dump. If there is any other error Msg then the transaction dump will fail and email will be send. At the end, an status of the transaction log dumps is email to the DBA
EcCoDbSyb_SedFile∞	This file contains all the database that don't need to be dump (i.e., temp, model, etc.)
EcCoDbSyb_DboMail	This file contains the email list of all the DBA's.
EcCoDbSyb_DbStat∞	This script updates the index table of a database. This script is called from EcCoDbSyb_DumpDb after each successfully database dump.
EcCoDbSyb_CkErrorLog∞	This script checks for specific database error messages from the Sybase Error Log File every hour and emails the error messages

	to the DBA's in the EcCoDbSyb_DboMailfile.
EcCoDbSyb_tran_log.awk	This script matches the current hour with the hour the error messages were enerated in the Error Log File. If errors found, the messages are saved in a mailfile and sent to DBA's.

THE FOLLOWING FILES MUST BE MODIFIED BEFORE RUNNING ANY OF THE ABOVE SCRIPTS:

EcCoDbSyb_SetupKsh∞	Make user you have the SYBASE files under /usr/ecs/OPS/COTS/sybase
EcCoDbSyb_SedFile	Add any other database that might not need to be backed up. The databases that are listed in this file do not need to be backed up.
EcCoDbSyb_DboMail∞	Add/delete the email of the DBA and any other email that might need to be added/deleted. All the errors and status will be send to them.
run_sybcron∞	The following is an example on the crontab file that should be run by a sybase user. The first one will run the EcCoDbSyb_DumpDb script that dumps the databases at midnight from Monday to Saturday. The second one, EcCoDbSyb_DumpTran script that dumps the transaction logs will run tree times a day, 10AM, 1PM and 4PM from Monday to Saturday. The Third one, EcCoDbSyb_CkErrorLog that check the SYBASE error log file will run every hour from Monday to Saturday.

0 0 * * 1-6 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_DumpDb

0 10,13,16 * * 1-6 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_DumpTran

0 * ** 1-6 /usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin/EcCoDbSyb_CkErrorLog

NOTE: Make sure there is an OPS mode directory with all script files.

All these scripts reside in "/usr/ecs/OPS/CUSTOM/dbms/COM/DBAdmin" directory. The assigned site DBA will be responsible for maintaining, modifying and applying necessary changes that are applicable to their site as for (security, and backup schedule).

SQL Server backups are performed nightly by a **cron** job which runs the **run_sybcron** program located in the **$SYBASE**/ directory. The following table of definitions will be used throughout the rest of this section.

Table 4.10-2. Automatic Backup Components

Component Name	Function(s)
run_sybcron	File added with the crontab -e command, contains several executable cron commands. **Example:** 00 19 * * 1-6 /data1/COTS/sybase/sybasedump
EcCoDbSyb_DumpDb	Controlling script that performs the following functions: run isql to create the Backup Statements run isql to execute the Backup Statements record the results of the Backup Statements in Backup Files copy the Backup Files to the Backup Subdirectory create the Backup Summary "greps" successful Dump statements along with any errors generated, sends e-mail to the DBA and writes them to the backup_summary file
sp	SQL Server password file - contains password for backup role

No intervention in the Automatic Backup Process is required by the DBA, though periodic checks of the Backup Subdirectories are recommended.

4.10.2 Manual Backups

Manual backups can be performed at any time by the System Administrator and are recommended for the following situations:

Any change to the **master** database - this includes new logins, devices, and databases

Any major change to user databases - a large ingest or deletion of data, definition of indexes

Other mission-critical activities - as defined by the DAAC Operations Supervisor.

Both the **dump database** and **dump transaction** command processing are off-loaded to the backup server, and will not affect normal operations of the database. These commands are performed by the System Administrator on appropriate databases as follows:

Syntax:

1> dump database master to
"/usr/ecs/OPS/COTS/sybase/sybase_dumps/dumps/dbname.dat.MMDDHHMM."

go

After dumping the database, compress the dump file by executing:

%compress
/usr/ecs/OPS/COTS/sybase/sybase_dumps/dumps/dbname.dat.MMDDHHMM.

Syntax:

dump transaction pdps_db_ops to
“/usr/ecs/OPS/COTS/sybase/sybase_dumps/trans/pdps_OPS.tran.YYMMDDHHMM"”

go

4.10.3 Manual Recovery

Manual recovery of a user database is performed by the System Administrator by the use of the **load database** and **load transaction** commands. For issues concerning the **master** database, please consult your System Administrator’s Guide for assistance. It is recommended that any user database to be recovered be dropped and created with the **for load** option., The **databasename**.sql along with any **alter.databasename**.sql scripts can be , combined into one script which will re-create the user database with the **for load** option. This will insure the success of the **load database** and **load transaction** commands.

4.10.4 The BulkCopy Utility

The **bcp** utility is located in the **$SYBASE**/bin directory and is designed to copy data to and from SQL Server databases to operating system files.

4.10.4.1 Requirements for Using bcp

In general, you must supply the following information for transferring data to and from SQL Server:

Name of the database and table

Name of the operating system file

Direction of the transfer (in or out)

In order to use **bcp**, you must have a SQL Server account and the appropriate permissions on the database tables and operating system files that you will use. To copy data **in**to a table, you must have **insert** permission on that table. To copy data **out** to an operating system file, you must have select permission on the following tables:

The table being copied

sysobjects

syscolumns

sysindexes

bcp Syntax

bcp [[database_name].owner.]table_name {in | out} datafile [-e errfile] [-n] [-c]

[-t field_terminator] [-r row_terminator] [-U username] [-S server]

4.10.4.2 Example of User Database Recovery

The database **UserDB** was created using the following script excerpt: (stored in home/scripts/create.databases/userdb.sql)

create database UserDB on data_dev1 = 100 log on tx_log1 = 50 [with override]

and was modified using the following script excerpt: (home/scripts/create.databases/alteruserdb.sql)

Alter database **UserDB** on data_dev1=50

For the purposes of this example, the full database backup and transaction log dumps were successful and located in /usr/ecs/OPS/COTS/UserDB.dat and UserDB_tx.dat

1 . In the **$SYBASE**/scripts/create.databases directory, DBA makes a script file from the template.

Syntax: % cd /usr/ecs/OPS/COTS/sybase/scripts/create.databases

% cp template.sql userdb_for_load.sql

2 . Appropriate items are modified so that the script file resembles the following:

1> create database UserDB on data_dev2=100 log on tx_log2=50 **for load**

2> go

3> alter database UserDB on data_dev3=50

4> go

3 . DBA saves the script in $SYBASE/scripts/create.databases/userdb_for_load.sql

4 DBA runs the script from the UNIX command prompt.

Syntax: %isql -Usa -S**servername** -iuserdb_for_load.sql -ouserdb_for_load.out

5 DBA checks the userdb_for_load.out file for success

6 DBA loads the database from the full backup.

Syntax: 1> load database UserDB from

"/usr/ecs/OPS/COTS/sybase/sybase_dumps/week1/dbname.dat.MMDDHHMM"

go

7 DBA loads the transaction file from the transaction file dump.

Syntax: 1> load transaction UserDB from

"/usr/ecs/OPS/COTS/sybase/sybase_dumps/week1/dbname.tran.MMDDHHMM"

3> go

4.11 Database Performance and Tuning

Once your application is up and running, the DBA monitors its performance, and may want to customize and fine-tune it. Use the following software tools provided by SQL Server:

Setting query processing options with the **set** command

Setting database options with **sp_dboption**

Monitoring SQL Server activity with **sp_monitor**

Using **update statistics** to ensure that SQL Server makes the best use of existing indexes

Changing system variables using **sp_configure** and the **reconfigure** command

Placing objects on segments to spread i/o, improve throughput, etc. as described in section 4.4.4

For a complete discussion of issues related to SQL Server performance and tuning, refer to your SYBASE SQL Server Performance and Tuning.

4.12 Installation of the Applications

DBA should have physical devices configured before installing either autosys or remedy. Both applications use Sybase as their database.

4.12.1 Installation of the Application Database

The installation of the application databases has been automated using ECS Assistant. The application databases are created using the DbBuild script which can only be invoked through ECS Assistant or the Command Line. Scripts that ECS Assistant invokes are:

DbBuild - Create new empty database and loads with initial data

DbPatch - Upgrade to new schema while retaining existing data.

4.12.2 The AUTOSYS Application and other Configuration Issues

The AUTOSYS application works in tandem with PDPS/DPSs to schedule the jobs that run on Science Processor. Autosys installation is performed in /usr/ecs/OPS/COTS by the auto install program located in the autosys/install directory. The results of the installation are stored in an autosys_install.scr file located in the AUTOSYS home directory (/use/ecs/OPS/COTS/autosys). For pdps to run properly with AUTOSYS, the following activities are completed:

A user is defined named **autosys**

autosys user is added to the pdps database (OPS mode)

The autosys server is added to the sysservers table with **sp_addserver**

The server is added to the sysservers table on the AUTOSYS server with **sp_addserver**

4.12.3 Spatial Query Server (SQS)

SQS is a multi-threaded, Sybase Open Query database engine, which is required by the Science Data Subsystem (SDSRV). This product allows definition of spatial data types, spatial operators, and spatial indexing. SQS communicates with Sybase SQL Server to process SDSRV requests to push and pull metadata. SDSRV database server resides on an SGI machine. SQS also, reside on the same machine as SDSRV Sybase SQL Server.

Named X1acg01 - where X is the DAAC specific identifying character.

pathname - /usr/ecs/OPS/COTS/sqs222/bin/sqsserver

Should have one dedicated CPU per instance running. Defaults to one instance now, but may require additional instances later for performance reasons.

Requires one entry in the Sybase "interfaces" file per instance of the SQS server to be run.

Consult startup scripts in /etc/init.d/sybase and /etc/init.d/sqs_222

SQS requires a Sybase login with SA or sa_role and associated password to start. SQS environment variables requirements:

SYBASE = Location of the Sybase home directory. Example: /tools/sybOCv(TBD)

PATH = Must include in this order - /usr/bin; /usr/sbin;$SYBASE/bin

DSQUERY = Name of SQL Server to which to connect. From the $SYBASE/intefaces file. Examples - g0acg01 _srvr

DSLISTEN = Name of SQS server to use. Example - g0acg01_ srvr

SQSUSER = Name of the user (SA or sa_role) for system connection.

SQSPASSWORD = Password for the system connection login

The SQS startup script requires the following information:

SQSHOME = location of sqsserver binaries.

The following is a list of options that can be imbedded in the startup script, these options are beneficial, but they are not required.

4.12.3.1 SQS STARTUP OPTIONS:

-e path of the SQS server logfile. Example /usr/ecs/OPS/COTS/sqs222/sqs/bin/sqs_222.log

-u number of concurrent SQS connections. Recommend minimum of 125. Example -u 125

Usually started with a delay, after the SQL Server is started. This delay be sufficient for the SQL server to recover and come-up.

$SQSHOME/bin/sqsserver -e $SQSHOME/sqs_222.log -u $USER &

SQS has dependencies on Sybase, such as:

Sybase must be running prior to starting SQS

SQS user id that starts SQS, which is different from the application user ID must have admin privileges

SQS opens a connection to Sybase's because it writes to the Sybase System tables

SQS server thread runs under the userid sa. In order to avoid confusion when monitoring this thread, it is best to:

create a separate login and userid specifically to monitor SQS

grant sa_role authority to the userid created to monitor SQS

EXAMPLE: 1> **sp_adduser** sqs_mon

grant sa_role to sqs_mon

go

4.13 Configuration of XLV partitions for Sybase partitions .

In order to successfully convert the raw partitions into XLV, the following steps are to be strictly adhered to.

BACKUP all databases.

BCP the syslogins table from master.

bcp master..syslogins out file.out -Usa -P -c

bcp master..sysloginroles out file.out -Usa -P -c

Save all information regarding the sysdevices and database options by executing the following command:

sp_helpdb db_name -- each database

sp_helpdevice

sp_helpdb

Shutdown the backup and sql servers.

After the partitions have been updated to XLV, chown the sybase disks to sybase:users. Also, change the /etc/init.d/sybase script.

Cleanup some old sybase files from $SYBASE/devices and $SYBASE/install.

Execute sybinit to initialize a NEW sql server/backup (same server name).

Initialize sybsystemprocs database, as well.

Change sa password. (sp_password NULL,newpass)

Up the number of devices. (sp_configure "number of devices",20)

alter the database for master and tempdb.

Change sysservers to name the server, as well as the new backup server.

sp_addserver nodename_srvr,local,nodename_srvr

sp_addserver SYB_BACKUP,local,nodename_backup

execute scripts to initialize the devices and to create databases (for load).

load database dumps. Verify that database data and log are not on same device. If on same device, update sysusages by deleting the log that uses he data device and then update the size of the data device to the full size (plus the one used by the log).

uncompress the load file, manually.

online database.

BCP in the syslogins table.

BCP in the sysloginroles table.

Change the DB dbo.

Reset all DB options.

Dump all databases.

4.13.1 Backout Procedure

This is an addendum to the technical directive that was previously sent out, dated September 28,1999. The backout procedure is executed only if it is necessary to restore the old SQL server because of a XLV conversion failure. Please give me a call if you have any questions in following the procedures.

After restoring the sybase raw disk partitions from XLV partitions (normally done by the System Administrators), change the ownership of the devices to sybase:user. This step has to be executed by someone with root access.

Login as sybase, and cd to the $SYBASE/install directory.

Execute sybinit and configure a new sql server. Use the same configurations that were used from original sql server, ie. Port numbers.

Once you have configured a new server, get in to isql and execute the following:

change sa password

alter database master to same size as before

modify the sysservers with the sql and backup names

shutdown sql server and bring it up in single user mode (-m option)

uncompress the master database dump (do not load at this time)

cd to /usr/ecs/OPS/COTS/sybase/devices and copy the old sybsystemprocs.dat to the new one

load the dump of the master database (after the load, it would automatically shut down the sql server, so, start it up as normal and not as single user)

Now, at this point, all your databases should recover. If not, your databases would be marked as suspect. Then, check your devices and load each database from the dumps. Just follow the following steps:

Check all the devices and make sure that they do exist

Drop the databases using dbcc repair (dbname,dropdb) option

Re-create the databases with a "for load" option

Load each of the databases and do an online command after

Verify that your db options were set properly, as well as all the users on each database

Once, everything is back to its original state, do a complete backup of all your databases!

This completes the backout process.

4.14 Passwords Security

Security has become a sensitive issue throughout the IT Industry. The ECS program is also concern about security and the risks associated with security. As a result the following directive is issued to all DAACs.

All System Administrators and Database Administrators at the sites are responsible for easonable security measures when installing ECS custom software. This means:

Changing the permissions of online secure files to the minimum level required .

Backing up secure file(s) to removable media (floppy or tape) and removal of secure files immediately after installation is complete.

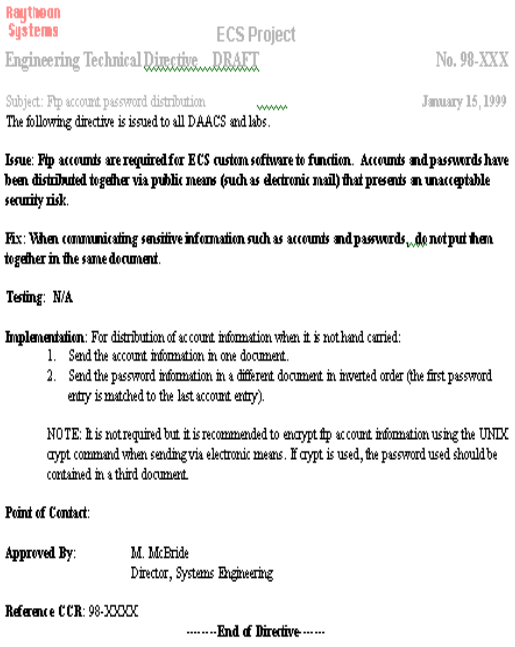
1. The media should then be kept in a secure location.

The following file is affect as result of this requirement on the ECS program.

A. /usr/ecs/<MODE>/CUSTOM/dbms/<SUBSYSTEM>/Ec<server>SybaseLogins.sql

B. Set permissions to 711 (user read, write, execute, group and other read only)

Figures 4.X-1 and 4.x.-2 are the Technical Directives issued by the Director of Systems Engineering.



Raytheon Systems

ECS Project

Engineering Technical Directive DRAFT No. 98-XXX

Subject: Ftp account password distribution January 15, 1999

The following directive is issued to all DAACS and labs.

Issue: Ftp accounts are required for ECS custom software to function. Accounts and passwords have been distributed together via public means (such as electronic mail) that presents an unacceptable security risk.

Fix: When communicating sensitive information such as accounts and passwords, do not put them together in the same document.

Testing: N/A

Implementation: For distribution of account information when it is not hand carried:

1. Send the account information in one document.
2. Send the password information in a different document in inverted order (the first password entry is matched to the last account entry).

NOTE: It is not required but it is recommended to encrypt ftp account information using the UNIX crypt command when sending via electronic means. If crypt is used, the password used should be contained in a third document.

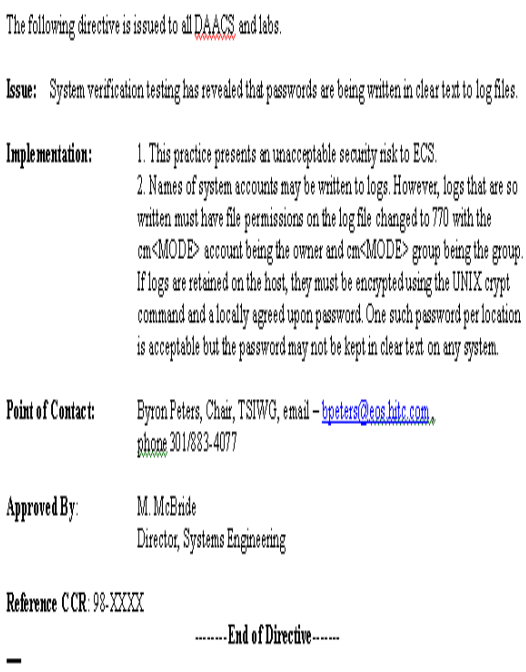
Point of Contact:

Approved By: M. McBride
Director, Systems Engineering

Reference CCR: 98-XXXX

-------End of Directive-------

Figure 4.x-1 Technical Directive



The following directive is issued to all DAACS and labs.

Issue: System verification testing has revealed that passwords are being written in clear text to log files.

Implementation:
1. This practice presents an unacceptable security risk to ECS.
2. Names of system accounts may be written to logs. However, logs that are so written must have file permissions on the log file changed to 770 with the cm<MODE> account being the owner and cm<MODE> group being the group. If logs are retained on the host, they must be encrypted using the UNIX crypt command and a locally agreed upon password. One such password per location is acceptable but the password may not be kept in clear text on any system.

Point of Contact: Byron Peters, Chair, TSIWG, email – bpeters@eos.hitc.com, phone 301/883-4077

Approved By: M. McBride
Director, Systems Engineering

Reference CCR: 98-XXXX

-------End of Directive-------

Figure 4.x-2 Technical Directive

4.15 Database NCR/TroubleTicket WorkArounds

4.15.1 OPS:5A.03:EGS-10:No graceful way of stopping MODAPS Ingest

Ticket-Id : SMC000000001859

Unique-Identifier : EDC000000003826

Long-Description : In order to support periodical Sybase backups along with any other planned system shutdown, OPS needs a procedure for gracefully shutting down MODAPS ingest that doesn't cause us to lose any data.

Problem Type : Software

Related CCR : NCR 25302

Resolution Log (End User Sees) :

Short Description : OPS:5A.03:EGS-10:No graceful way of stopping MODAPS Ingest

Software Information : Create

Software Resource :

Submitter Home DAAC : EDC

Submitter ID : bbates

Subject: Ingest shutdown

Author**: Bryn Ardanuy at 11-ECS-2**

Date: 1/13/00 2:47 PM

Cheryl,

In order to shut down Ingest gracefully, first the front ends needs to be shut down - Auto and Polling. And also, don't start any new media ingests from the GUI. Then once the requests have all completed, the Request Manager and Granule Servers can be shut down.

If there is not enough time to do a graceful shutdown, then all of the Ingest servers can be brought down. When they are warm started, the requests which were active before the shutdown, should restart and go to

completion.

4.15.2 PLS: PLS should not submit more than 1 subscription for one ESDT

NCR ID: ECSed24368 Status: ASSIGNED-IMPLEMENT Submitted: 991007
NCR Class: OPERATIONS Project: OPS_Documentation Enclosures: 7

NCR TITLE...
PLS: PLS should not submit more than 1 subscription for one ESDT

PROBLEM INFORMATION... ANALYSIS INFORMATION...
Build Name[*]: Drop 5A.03 Evaluate Engineer: xihaihu
Test Site[*]: GSFC DAAC Assigned To: rodney
Detection Method[*]: Acceptance Test Analysis Due Date (yymmdd):
Detected-In-Phase[*]: Integration
Test Case ID: COST ANALYSIS INFORMATION...
Machine Name: Cost[*]: MEDIUM
Severity (1=Showstopper)[*]: 2 Estimated Fix Time(hrs): 20
Mission-Criticality (Admin Use Only): 0 Estimated Fix Date(yymmdd): 991015
Mode [*]: OPS Workaround Available? Y
Trouble Ticket:
DAAC Trouble Ticket: TO BE FIXED INFORMATION...
SUBMITTER INFORMATION... Problem type[*]: source code
Submitter: xihaihu Recommended change[*]: source code
Implement Due Date (yymmdd):

Page 1/2

NCR Class: OPERATIONS Project: OPS_Documentation Forwarded 991209

FORWARDING INFORMATION...
Old class: OPERATIONS
Old project: OPS_PLS

Forwarded by: Shiela Barriere

Page 2/2

********** **Problem** **********
Please describe the problem you are experiencing below, including
what you did, what you expected to happen, and what actually happened:

During MOSSIII test at GSFC, it was found SubMgr has multiple subscriptions on insert events for one ESDT. This force SubMgr to process one granule multiple times which slows down the system significantly.

********** **Analysis** **********
Please describe your Analysis:

When it comes to subscribe to an insert event for a certain ESDT for SubMgr, PRE first look up PDPS DB, where a subscription ID is stored if the subscription has been placed previously. If a subscription ID is found, PRE will use it to confirm is validity with subscription server, if it gets confirmed, it will not submit the subscription again. However, if PDPS DB gets reset, the subscription ID is lost from PDPS DB, and PRE will submit subscription without checking.

We need to prevent this from happening by checking existing subscriptions in subscription server database based on ESDT only(because we don't have the subscription ID).

Based on conversation with Dawei, subscription server does not have existing interface to check subscription based on ESDT and userID. CSS needs to add interface in order to make this work.

********** **Workaround** **********
Please describe the workaround to the problem:

Removed extra subscriptions for SubMgr using subscription server GUI.

********** **Op Procedure (Added 991115 by xihaihu**) **********
The operational procedure to correctly "reset"
PDPS database is:

"Whenever PDPS database is reset(not including database cleanup by running EcPlDbClean script), the operator is

also responsible to remove all the PLS subscriptions
in CSS subscription server database(subscriber
ID is SubsMgr). This can be done through running
"EcSbSubServerGUI".

NOTE: AFTER PDPS DATABASE IS RESET, AND THE ABOVE
PROCEDURE IS FOLLOWED, NO SUBSCRIPTION NOTIFICATION
WILL COME THROUGH FOR EXISTING JOBS IN THE NEWLY
LOADED DATABASE. ONLY NEW JOBS GENERATED
BY PRE WILL WORK NORMALLY REGARDING SUBSCRIPTIONS."

By following this procedure, the problem described
in this NCR will not happen.

********** Resolution **********
Please describe your resolution to this problem:

Document the enclosed "Op Procedure" for PDPS database
reset.

********** Previous Merge Build IDs **********
Please enter any related Merge Build ID(s) information:

********** Back to R (Added 991209 by sbarrier) **********
The NCR Board moved this back to R at the T scrub on 12/9/99 so that the
documentation aspects can be addressed.

********** History **********
xddts 991007 123558 Submitted to OPS_PLS by xihaihu
xddts 991007 123725 Enclosure "Problem" added by xihaihu
xddts 991007 123732 N -> A (Assign-Eval to xihaihu) by xihaihu
xddts 991007 123831 Enclosure "Analysis" added by xihaihu
xddts 991007 123853 Enclosure "Workaround" added by xihaihu
xddts 991007 123853 A -> B (Cost-Analyzed) by xihaihu
xddts 991007 124537 enclosure "Analysis" edited by xihaihu
xddts 991007 124548 B -> R (Assign-Implement) by xihaihu

xddts 991007 130622 enclosure "Analysis" edited by xihaihu
xddts 991115 134030 Enclosure "Op Procedure" added by xihaihu
xddts 991116 095822 enclosure "Op Procedure" edited by xihaihu
xddts 991116 113401 Enclosure Resolution added by xihaihu
xddts 991116 113404 Enclosure Previous Merge Build IDs added by xihaihu
xddts 991116 113404 R -> M (Merged) by xihaihu
xddts 991116 113408 Fields modified by xihaihu
xddts 991116 113436 Fields modified by xihaihu
xddts 991116 113525 enclosure "Resolution" edited by xihaihu
xddts 991116 113536 M -> T (In-Test) by xihaihu
xddts 991209 112542 Forwarded from OPS_PLS to OPS_Documentation by sbarrier
xddts 991209 112550 T -> R (Assign-Implement) by sbarrier
xddts 991209 112554 Fields modified by sbarrier
xddts 991209 112638 Enclosure "Back to R" added by sbarrier

Bryn

4.15.3 Sybase BACKUP SCRIPTS

NCR ID: ECSed22290 Status: ASSIGNED-IMPLEMENT Submitted: 990421
NCR Class: OPERATIONS Project: OPS_DBDM Enclosures: 4

NCR TITLE...
Sybase BACKUP SCRIPTS - TO BE FORWARDED TO

PROBLEM INFORMATION... ANALYSIS INFORMATION...
Build Name[*]: Drop 4PY Evaluate Engineer: mmuganda
Test Site[*]: EDC DAAC Assigned To: hcolglaz
Detection Method[*]: Customer Use Analysis Due Date (yymmdd):
Detected-In-Phase[*]: DAAC Activity
Test Case ID: COST ANALYSIS INFORMATION...
Machine Name: Cost[*]: LOW
Severity (1=Showstopper)[*]: 3 Estimated Fix Time(hrs): 40
Mission-Criticality (Admin Use Only): 0 Estimated Fix Date(yymmdd): 990831
Mode [*]: TS1 Workaround Available? Y
Trouble Ticket: SMC000000000616
DAAC Trouble Ticket: EDC000000001301 TO BE FIXED INFORMATION...
SUBMITTER INFORMATION... Problem type[*]: procedure
Submitter: langevin Recommended change[*]: procedure
Implement Due Date (yymmdd):

Page 1/2

NCR Class: OPERATIONS Project: OPS_DBDM

********** Problem **********
Long Description :
Backups do not occur if there is activity in the
database at the precise time that the backup kicks off.
Mailer daemon 04/19/99 18:56:35

ematthew 04/21/99 07:30:33
Sorry about this, we forwarded a bunch of tickets
-Tammy SMC616 Ticket-Id : EDC000000001301
Detailed Resolution Log: 04/19/99 12:39:45 langevin. If there is an insert/up
date/delete happening at the precise time the databases go to dump their transa
ction logs, or to dump the whole database, it will not do this if it is in use.
This means that nightly dumps are not caught on the Legato Networker backups
because the dumps have not happened. This could be potentially dangerious for
production. Is there a way to script in the backup scripts for a recheck if the
dumps did occur and if not, to try again? This should be a high priority item
to be forwarded to Landover.
lf
m0msh03@@/home/ematthew> dangerous f

<< end of description >>

********** Analysis **********
Please describe your Analysis:

Backups fail when dbcc is attempted and their is activity in the database.
dbcc requires exclusive lock on the database tables. dbcc is needed to ensure
database integrity prior to a backup. EDC needs to incorporate a maintenance
window for doing database backups were there is no activity on the SQL server
so that dbcc.

DDM will work with EDC and all DAACs to establish a maintenance window.

Until maintenance window is established. DAAC dba should manually execute backup scripts for any database where error log shows that backup chron job failed.

********** Workaround (Modified 990614 by gsloan) **********
Please describe the workaround to the problem:

EDC DBA should manually execute backup script for and SQL server were error log indicates that backup chron job aborted.

********** Unacceptable workaround (Added 000121 by rockvam) **********
EDC DAAC management has found this workaround to be not
acceptable as it requires 3 4-hour down times per week.

John Daucsavage
EDC 1/21/00
Per G.Sloan

********** History **********
bugs 990317 000000 Submitted by Remedy via ddts@eos.hitc.com
xddts 990504 122219 N -> A (Assign-Eval to mmuganda) by mmuganda
xddts 990608 182236 Enclosure "Analysis" added by mmuganda
xddts 990608 182316 Enclosure "Workaround" added by mmuganda
xddts 990608 182316 A -> B (Cost-Analyzed) by mmuganda
xddts 990608 182353 B -> R (Assign-Implement) by mmuganda
batchbug 990614 141518 Fields modified by gsloan
batchbug 990614 141518 Enclosure "Workaround" modified by gsloan
batchbug 000121 161514 Enclosure "Unacceptable workaround" added by rockvam

4.16

TBD

This page intentionally left blank.

M&O Procedures:
Section 4B—Replication Server Administration

4. Replication Server Administration

ECS Sybase Replication Server Administration Overview

Sybase Replication Server will be used for the ECS Project starting in release 5B. Implementation of Replication Server for Earth Observing System Distributed Information System (EOSDIS) Core System (ECS) will require the support of the Management Subsystem (MSS). The deployment of the Sybase Replication software and it's support tools that distributes data across distributed active archive center (DAAC)s imposes additional operational requirements beyond original concept of a stand-alone DAAC operations. This new concept will allow all science users to register only once at their specified Home-DAAC and be able to request data from all DAAC sites.

In release 5B Sybase Replication Server's most basic model "primary copy model" be implemented. In this model, database transactions are replicated from the primary database (locate at the System Monitoring Center) to a replicate database via one or more replication servers. The model assumes 'ownership' of the data by the primary database. This means, the data can only be updated by clients (local DAACs) connected to the primary database and that clients connected to the replicate database have read-only access to the data.

The transactions are continuously replicated asynchronously from the primary to the replicate database. Replication is transparent to client applications that submit database transactions to the primary database. However, this feature creates a latent period (the time it takes to propagate the transaction across the network) during which data in the primary and replicate databases are inconsistent with each other. Client programs using these databases must account for this latency.

Sybase Replication Server implements this model using replication definitions for tables at the primary database and subscriptions for tables at the replication definitions. The replication definitions specify the location of the primary data while the subscription specifies the location of the replicate data. The replication definition and subscriptions are specified at the table or stored procedure level. (only table replication is implemented for the release 5B) Replication does not require replicating all tables in a database.

The primary copy model can be viewed as a building block to create other models. The model attempts to prevent data inconsistencies that would be created by updating and replicating copies of the same data in two databases simultaneously. The model stipulates that inserts, updates, and deletes to the data can only occur at the primary database, while select statements may occur at either the primary or the replicate database. The model assumes the enforcement of data ownership using custom developed database triggers, client code, or operational procedures.

The deployment of software that distributes data across distributed active archive center (DAAC)s imposes additional operational requirements beyond stand-alone DAAC operations. These enterprise level requirements conflict with the requirements for autonomous DAAC operations in two major areas: 1) capabilities that rely on database replication will be temporarily disabled

between DAACs if the databases and/or software are at different schema/drop levels; 2) administration of replicated databases requires coordination between DAACs.

This document focuses on the operations and administration of a replication system in a multi-site configuration by examining several operation scenarios involved in replication software installation and Replication Server administration. Although Sybase Replication Server supports same-site replication for warm-standby or load balancing needs, this document will focus exclusively on the issues involved in administering Sybase Replication Server for cross-site data distribution.

Cross DAAC Primary Copy Model Components

Figure 1 illustrates the components used in a primary copy model that uses two replication servers. This example is for illustrative purposes only. The ECS implementation will be described in the next section.

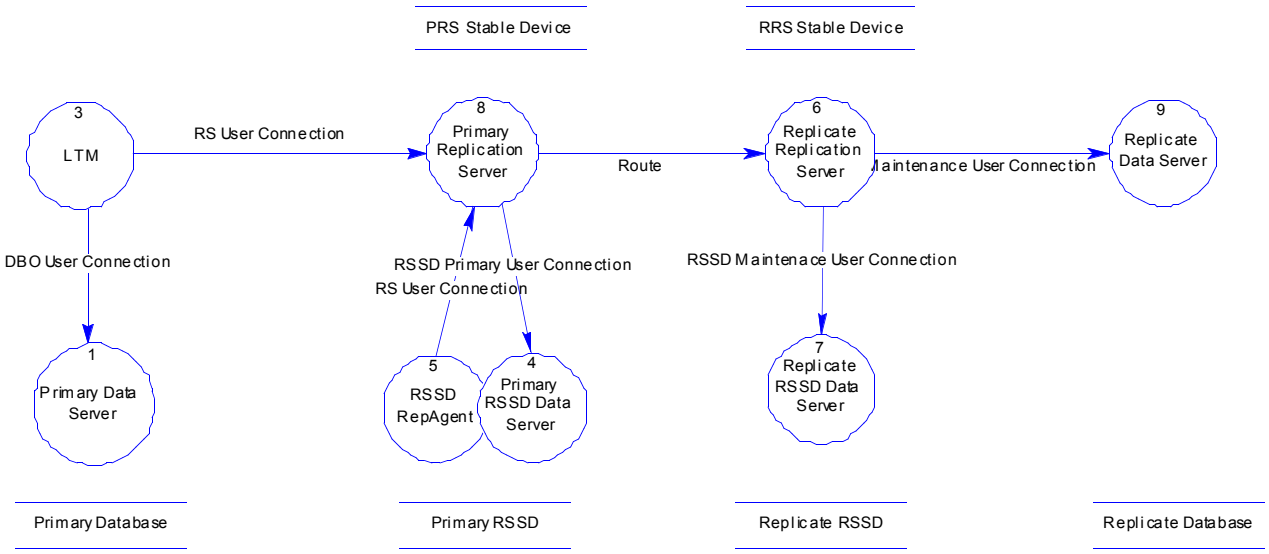


Figure 4B-1 Replication Server Components

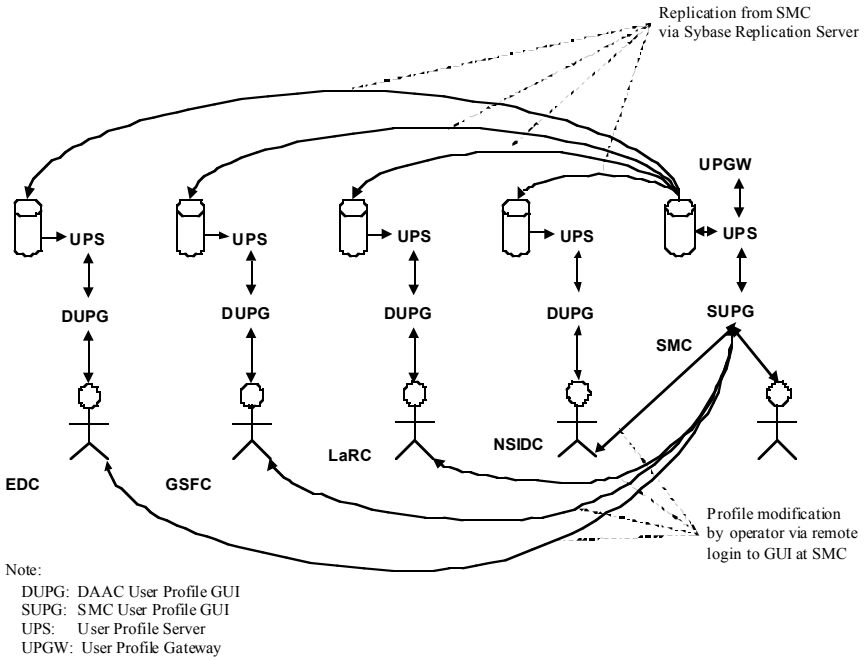
Components:

DAAC Component	Description
Primary Data Server	The primary data server is the Sybase SQL server that maintains the primary copy of data that is being replicated.
Primary Database	Contains the copy of data that can be updated by application programs.
LTM	The log transfer manager (LTM) is a Sybase Open Server application that transfers replicate database transactions to a primary replication server and moves the secondary truncation point in the primary database transaction log. The LTM connects to the primary data server as the primary database DBO and

	to the primary replication server as specified when the primary database is added to the domain.
Primary Replication Server	The primary replication server (PRS) is responsible for forwarding replicate database transactions to the replicate database. The PRS maintains connections to the replicate replication servers (route) and maintains a connection to its database, the RSSD.
Primary RSSD Data Server	The primary RSSD data server maintains the primary RSSD.
RSSD RepAgent	The RSSD RepAgent is a thread in the primary RSSD data server that transfers replicate RSSD database transactions to the PRS. The RSSD RepAgent connects to the PRS as specified when the PRS is added to the domain.
Primary RSSD Database	This database houses the information required by the replication servers to operate.
PRS Stable Device	The PRS stable Device contains a FIFO queue for each primary and replicate database. Transactions are transferred from a primary database queue to a replicate database queue after the LTM sends the transaction's commit. Once a transaction is moved to the replicate database queue, the primary replication server sends the transaction to the replicate replication server.
Replicate Replication Server	The replicate replication server (RRS) is a replication server that receives replicate transactions from a primary replication server and applies the transaction to a replicate database. The RRS maintains a maintenance user connection for each replicate database.
Replicate RSSD Data Server	This server houses the RSSD for the RRS.
Replicate RSSD	This database contains information that is required for the RRS to apply replicate database transactions to a replicate database.
RRS Stable Device	The RRS stable device is a file system that contains a FIFO queue for each replicate database. Replicate database transactions are pushed into the queue before being applied to the replicate database.
ReplicateData Server	This server houses the replicate database and is updated by the RRS.
Replicate Database	The database that contains the replicate data.

Release 5B Overview

The release 5B design implements several important changes from the original ECS concept. The first and most obvious change is that the System Monitoring Center SMC database becomes the primary database for all records. This means that all account request, profile creation, and profile modification activities must take place at the SMC. The second change is the use of Sybase replication which will automatically send database changes from the SMC to the other sites once they are committed, generally leading to a significant decrease from the time a change is made to the time in which that change is propagated to other sites. These changes also require modifications both to the database configurations and to custom code.



Drop 5B DAAC Operator User Profile Database Interactions

Figure 4B-2 Drop 5B DAAC Operator User Profile Data Interactions

GDS Order Tracking

The GDSGW will be located at System Monitoring Center (SMC) and will support Aster GDS submitted cross-DAAC product requests and product request status queries. Product requests will be routed to and processed by the appropriate DAAC. The DAAC filling the request will update a local copy of the request tracking data. The request tracking data will be replicated to EDC to support the product request status queries submitted via the GDSGW.

Since a product order may contain several requests, the order and request information stored at SMC will contain all requests processed at one or more DAACs. On the other hand, the DAAC processing the request will contain only its portion of the request(s) for the order.

This requirement imposes an inconsistency in the order data between sites, since the order data is a summation of requests. GDS order tracking has a requirement to preserving the order information,

independent of individual DAAC operations. The requirement stipulates that request delete transactions are **not** replicated back to EDC.

User Profile Distribution and User Registration Interaction Flow Diagram

Figure 4B-3 depicts the User Registration Interaction Flow as it is anticipated to be during the initial deployment of Release 5B. As the figure suggests, a science user who wishes to become a registered user associated with a particular DAAC (the Home DAAC) submits a New User Request (A.1). The user employs the User Registration Tool to submit the request to the User Registration Server at the SMC (A.2) The New User Request is retrieved by the User Services Representative at DAAC operations, logged in to the SMC remotely and using the Account Management tool (A.3). Note that the representative sees only requests specifying that representative's DAAC as Home DAAC are available to that representative for review. The representative reviews the information in the request and completes the User Profile (A.4). Upon completion of the User Profile at the SMC, the system sends confirmation to the user (A.5). With secure transmission of information, it is possible to send information electronically for printing of the User Profile information, including password, to the DAAC, so that the information can be printed and sent to the user by surface mail (A.6). The replication process (A.7) ensures that each DAAC has the entire User Profile database available on the local User Registration Server for viewing.

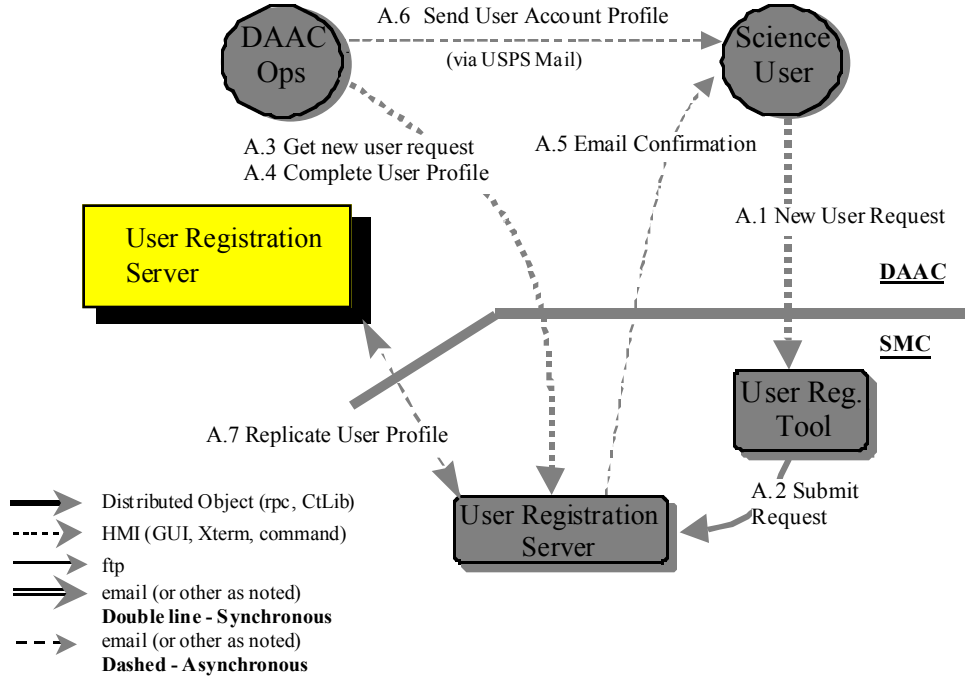


Figure 4B-3. User Registration Interaction Flow Diagram

User Registration Interaction Table - High-Level Operational View

Table 4B-1 provides the Interaction - High-Level Operational View: User Registration.

Table 1. Interaction Table - High-Level Operational View: User Registration

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
A.1	New User Request	Science User	User Registration Tool	None	None	Science user loads User Registration Tool, via its URL, from a favorite Web Browser. Science user fills out form with initial registration information. This information includes: user name, address, telephone number, email address and user verification key (for security confirmation). Request is queued at the SMC.
A.2	Submit Request	User Registration Tool	User Registration Server	None	None	User Registration Tool submits the new user's request. The request is queued at the SMC, awaiting the DAAC User Services staff from the user's selected home DAAC o confirm the new user.
A.3	Get New User Request	DAAC User Services Representative	User Registration Server	None	None	DAAC User Services Representative (periodically) checks for new user registration requests. In this case the request for our new user is found. User Services staff checks the information provided.
A.4	Complete User Profile	DAAC User Services Representative	User Registration Server	None	None	DAAC User Services Representative completes the new user's User Profile. The request is marked as confirmed and accepted. DAAC User Services Representative may call Science User for any further information or clarification. The Representative may authorize any special privileges (e.g., access to restricted granules, submission of ASTER Data Acquisition Requests) at this time.
A.5	Email Confirmation	User Registration Server	Science User	None	None	User Registration Server e-mails confirmation of the user's registration request.
A.6	Send User Account Profile	DAAC User Services Representative	Science User	None	None	DAAC User Services Representative sends complete user account profile, including user name and password, to Science User via USPS mail.

(Continued)

Table 1. Interaction Table - High-Level Operational View: User Registration

Step	Event	Interface Client	Interface Provider	Data Issues	Step Preconditions	Description
A.7	Replicate User Profile	Sybase	Sybase	None	None	The user profile is replicated at each DAAC. Each DAAC is capable of browsing user profiles locally. User profiles can only be created or modified at the SMC by operators from the user's home DAAC (who logs in remotely) or by select SMC operators.

User Registration Component Interaction Table

Table 2 provides the Component Interaction: User Registration. The information in this table provides more detail on specific interactions among ECS components for each of the main elements identified in Table 1 for those interested in what is occurring between ECS software configuration items and their components during user registration processes.

Table 2. Component Interaction Table: User Registration

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.1.1	Startup User Registration Tool	Science User	EcClDtUserProfile Gateway	Web Browser	Science User invokes the configured Web Browser with the URL of the User Registration Tool.
A.1.2	Input User Registration Information	Science User	EcClDtUserProfile Gateway	Web Browser	The Science User populates forms with ECS registration information. This information includes: user name, address, telephone number, email address and user verification key (for security confirmation). The user then submits this information.
A.2.1	Submit User Registration Request	EcClDtUserProfile Gateway	EcMsAcRegUserSrvr	Distributed Object	The User Registration Tool submits the User Registration Request to the User Registration Server for approval.
A.2.2	Store a User Registration Request	EcMsAcRegUserSrvr	Sybase	CtLib	The User Registration Request is saved for approval by DAAC User Services.

Table 2. Component Interaction Table: User Registration (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.3.1	Startup User Registration Server GUI	DAAC Ops - User Services	EcMsAcRegUserGUI	Xterm	DAAC operations remotely start the SMC User Registration Server GUI after logging into the SMC.
A.3.2	Review New User Request	DAAC Ops - User Services	EcMsAcRegUserGUI	Xterm	On a periodic basis (based on DAAC policy), User Services checks for any new User Registration Requests.
A.3.3	Get New User Requests	EcMsAcRegUserGUI	EcMsAcRegUserSrvr	Distributed Object	Request all new User Registration Requests. The GUI connects to the Registration Server at the SMC.
A.3.4	Retrieve User Registration Requests	EcMsAcRegUserSrvr	Sybase	CtLib	All pending User Registration Requests for the operator's home DAAC are retrieved from the database.
A.4.1	Update User Request	EcMsAcRegUserGUI	EcMsAcRegUserSrvr	Distributed Object	DAAC User Services completes the User Profile from the request. Updated information includes V0Gateway User name, group and password.
A.4.2	Create User Profile	EcMsAcRegUserGUI	EcMsAcRegUserGUI	Distributed Object	User Registration Server takes the completed User Registration Request and makes a User Profile, registering the user.
A.4.3	Store a User Profile	EcMsAcRegUserSrvr	Sybase	CtLib	The User Profile is saved in the SMC User Profile Database.
A.5.1	Send E-mail	EcMsAcRegUserSrvr	CsEmMailRelA (to Science User)	e-mail	A Confirmation message is sent to the new ECS Science User, via CSS infrastructure mail services (CsEmMailRelA).
A.7.1	Replicate User Profile	Sybase	Sybase	EcMsRsDb	The user profile is replicated at each DAAC via the Sybase Replication Server.

User Registration Tools

For the initial Release 5B period, a user may access the web-based ECS User Registration Tool (URT) from a link on the EOSDIS Home page to apply for an ECS Account. Figure 4B-4 illustrates the top portion of the URT. Figure 4 illustrates the URT scrolled down to show the bottom portion of the window.

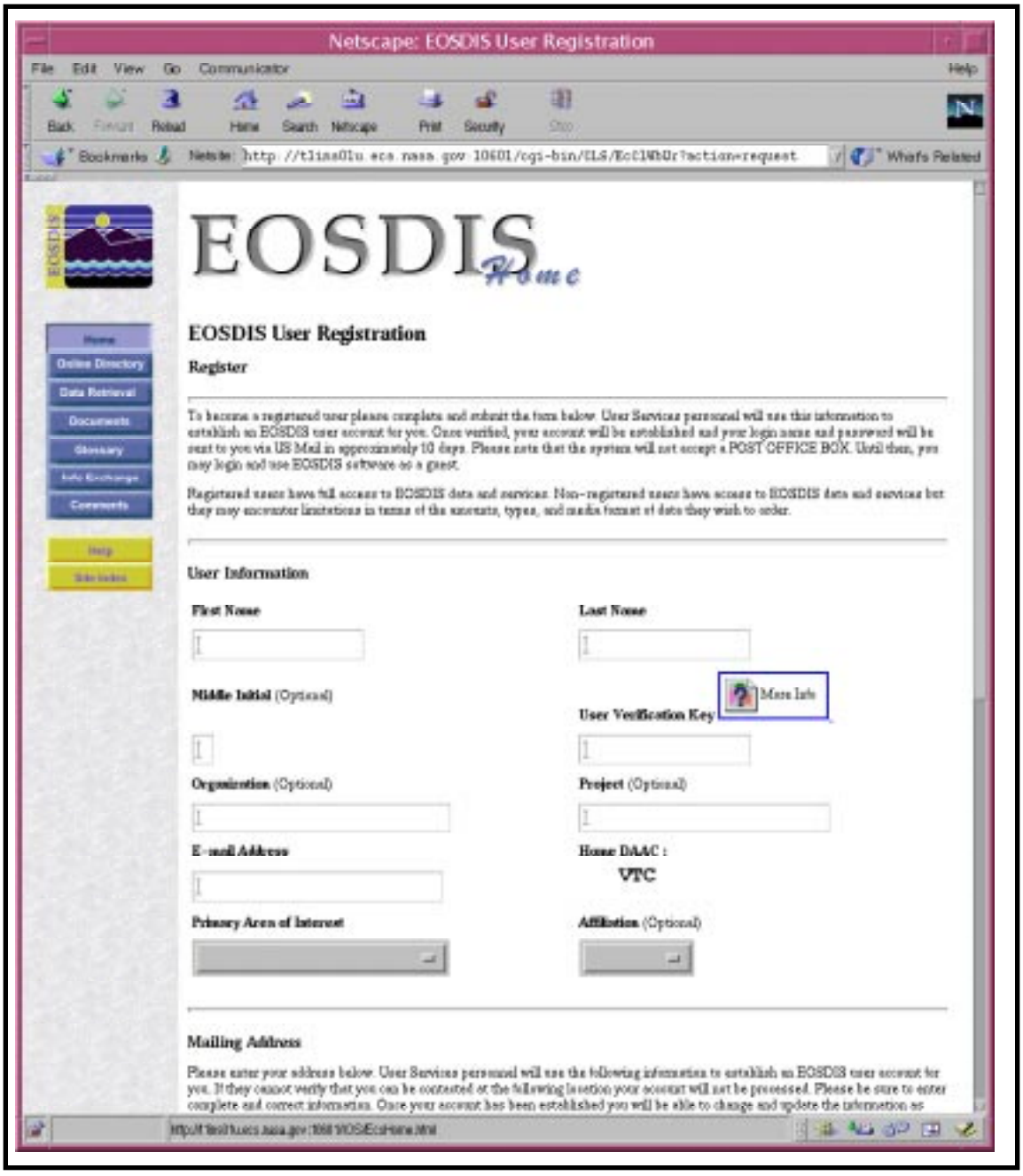


Figure 4B-4. User Registration Tool, Top Portion

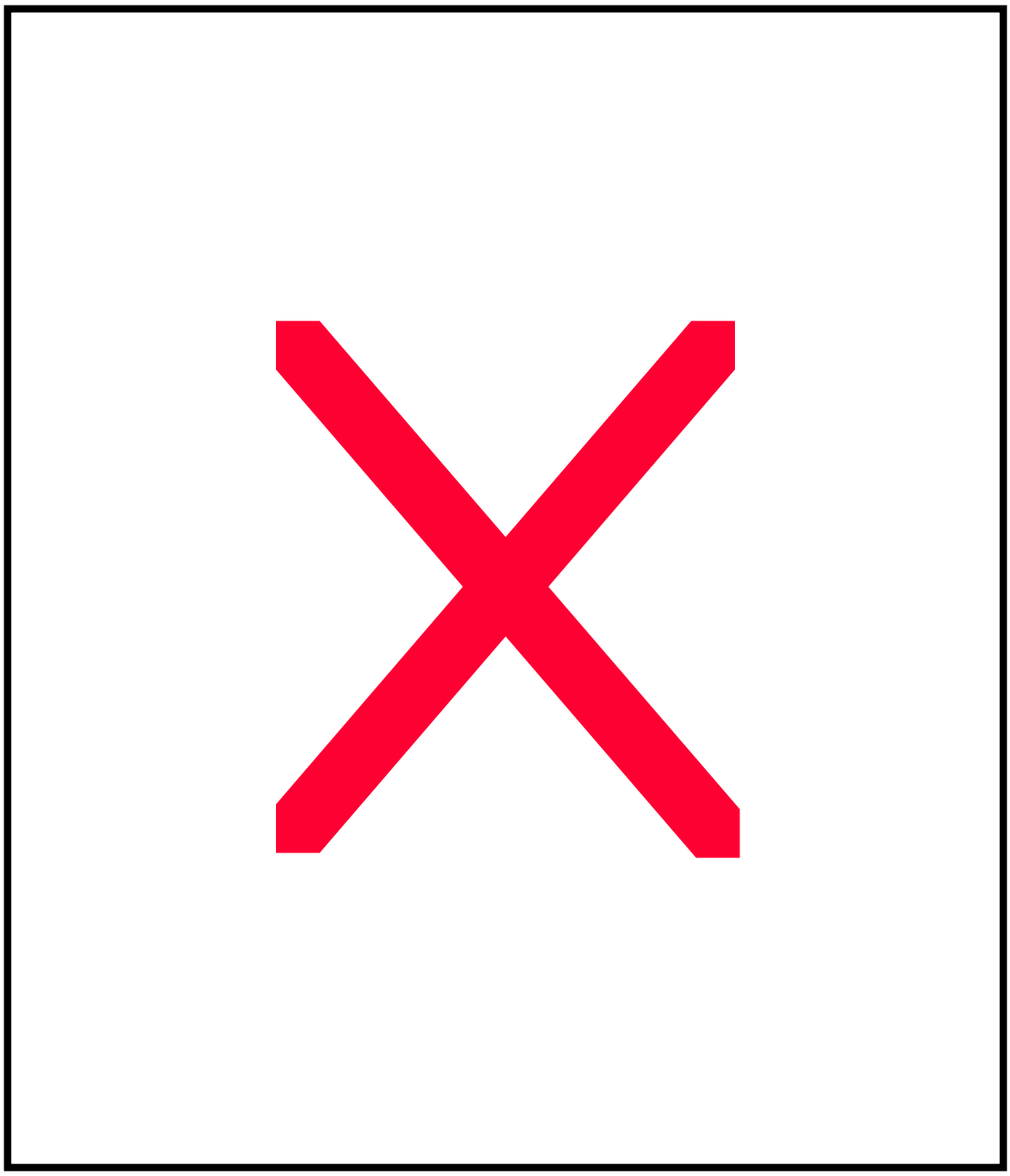


Figure 4B-4. URT Scrolled Down to Show the Bottom of the Window

The function of this URT will be supplanted by a patch later in the deployment of Release 5B that permits a user to apply for registration using the EOS Data Gateway Web Client. At that time, entering user profile data once will register a user for access to Version 0 products and services and application for an ECS account -- a "seamless" application to both communities.

At initial deployment of Release 5B, user registration is accomplished from the Request Account screen of the ECS User Account Management tool, accessed remotely at the SMC. Figure 4B-6 shows the Account Information tab of this screen.

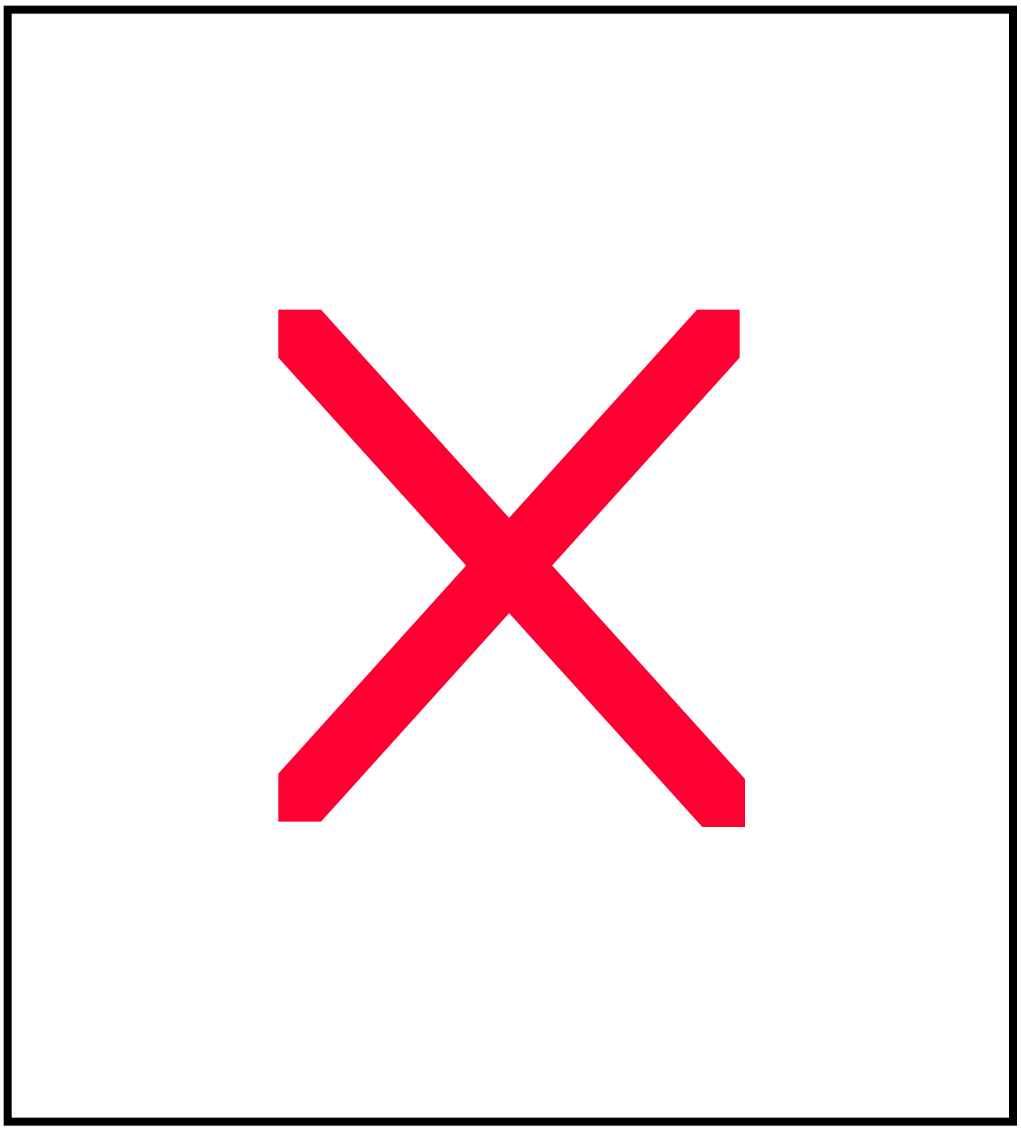


Figure 4B-6. Request Account, Account Information Tab (Note: Control buttons at the bottom of the screen appear only on the tool accessed at the SMC.)

The next block of data to be entered is user's personal information. If you have just entered the account information, the "**Request Account**" folder is still open. To add the user's personal information, you will need the "**Personal Information**" tab of this folder (Figure 4B-7). If the user has applied using the URT, these fields will already be populated.

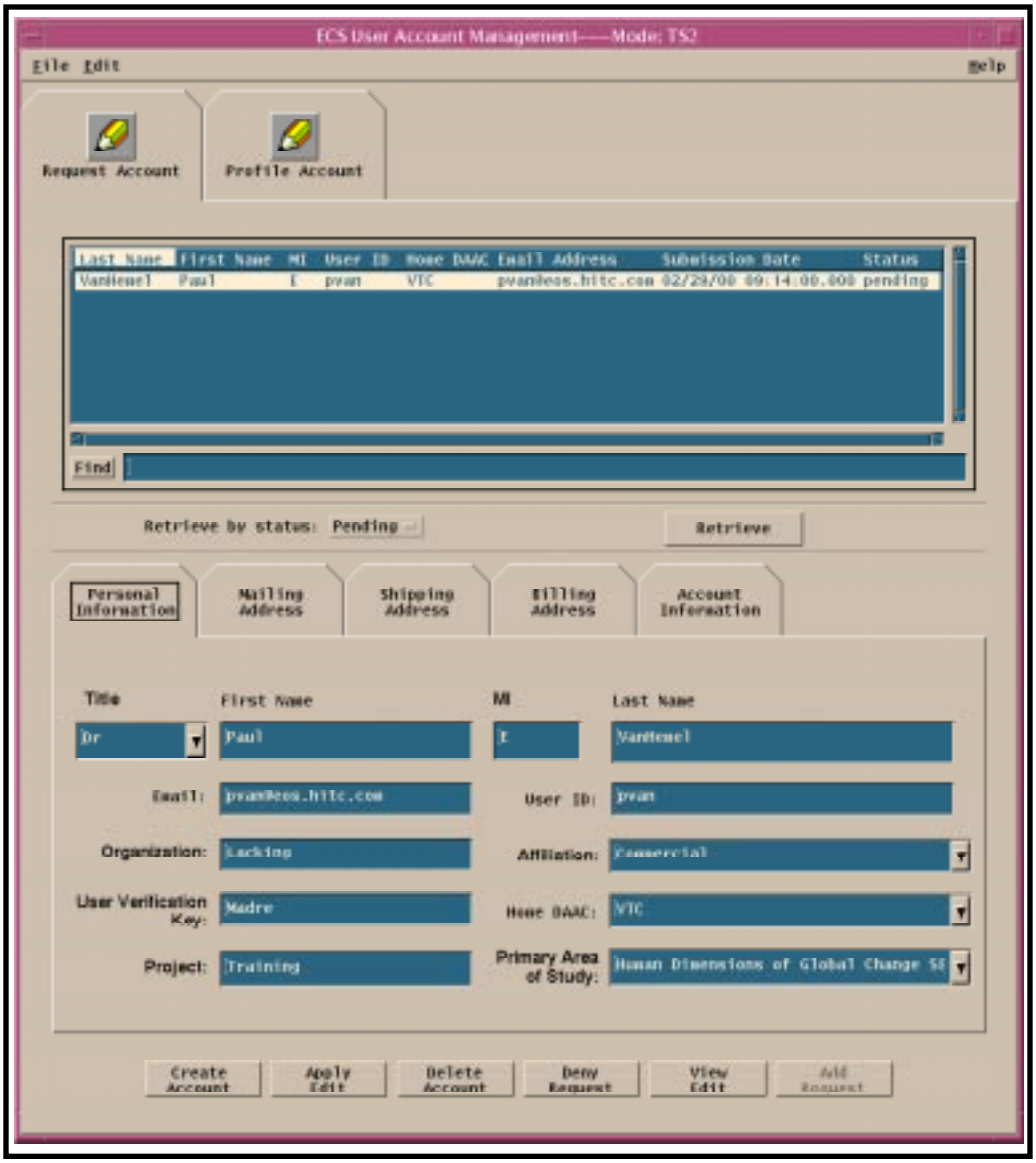


Figure 4B-7. Request Account, Personal Information Tab (Note: Control buttons at the bottom of the screen appear only on the tool accessed at the SMC.)

Mailing address, shipping address, and billing address information are entered using similar windows, or, if the user has applied for an account using the URT, these fields will already be populated.

To complete the creation of a new account, the pending account is selected and a click on the Create Account button results in the creation of the new account. It also results in automatic dispatch of an e-mail message to the user's e-mail address with notification that the account has been created. The User Services representative will complete the account registration process by providing the user with the initial ECS account password. The password dissemination is done in accordance with local DAAC policy.

Once the account is created, it will appear in the list accessible by clicking on the **Profile Account** tab, as illustrated in Figure 4B-8 showing the Personal Information and Account Information sub-tabs of this screen. (*Note*: The figure illustrates the screen with control buttons at the bottom. These buttons appear only on the Account Management tool at the SMC; the tool at your site does not include these buttons, but is otherwise the same.)

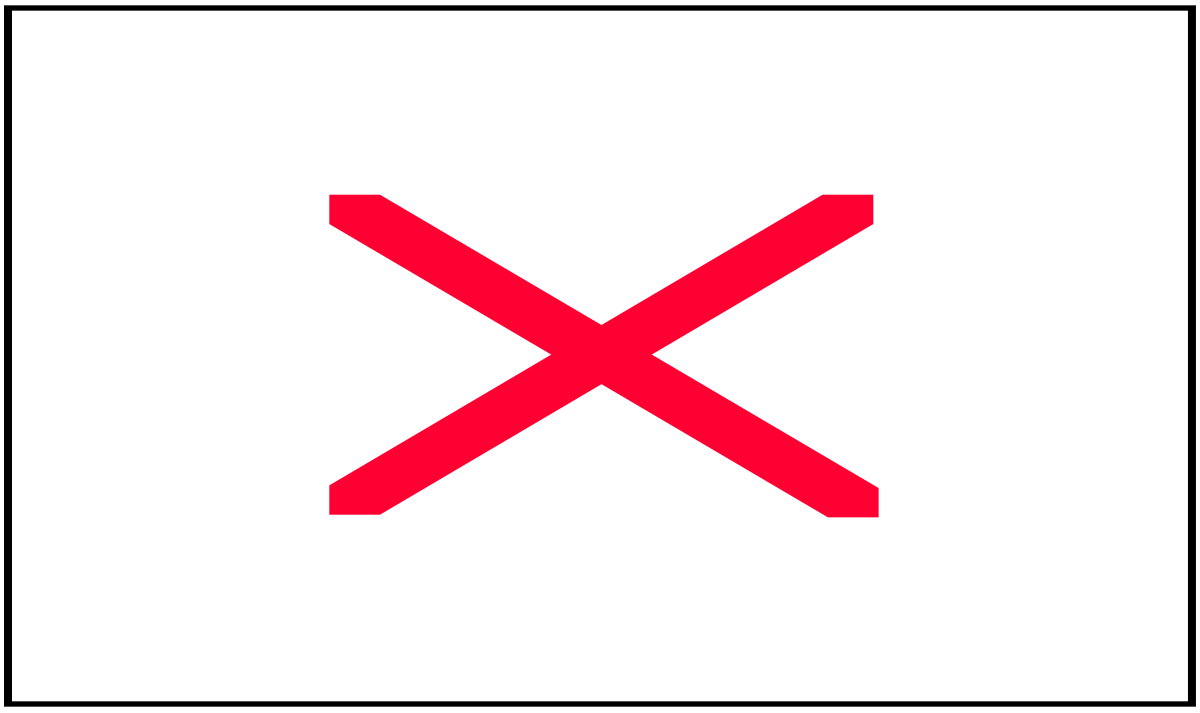


Figure 4B- 8. Profile Account Screens, Personal and Account Information (Note: Control buttons at the bottom of the screen do not appear on the local tool.)

The account can then be replicated to the User Profile databases at the DAACs, where it is accessible using the local Account Management GUIs. These local GUIs are read-only tools that do not display the control buttons for creating or modifying an account. If a user requires a change (e.g., new mailing address), the change is accomplished through the Account Management GUIs at the SMC, using secure remote access by the User Services representative.

Sybase Replication Server

The concept of a domain is useful when describing a replication system. Briefly, a domain is a set of replication servers and their associated components that communicate with each other. A domain can be one replication server that replicates data from a local primary database to another local replicate database (as in a warm standby application) or a domain can contain many replication servers distributed over a wide area network (WAN) as will be the case for the MSS.

Each domain requires one, and only one, ID server. An ID server is a replication server that is specified as such when it is installed. An ID Server assigns unique identifiers to domain components. The ID server must be the first replication server installed in a domain and must be accessible when any component is added to the domain.

When a replication server is installed (including the ID Server), the following components are created:

- a database called the replication server system database (RSSD) (the data server housing the RSSD must already exist)
- a stable device (queue)
- an interface (connection) to the RSSD data server
- a RepAgent for the RSSD

The RSSD contains system tables that are used by the replication server. In a mutli-server domain that implements consolidated distributed primary fragments, the RSSDs must also be replicated. The RSSD contains information about each domain component, component login ids and passwords, application specific objects such as replication definitions, replicate transaction identifiers, routes and connections, and replicate transaction errors.

The RSSD data model is documented in the manual *Replication Server Reference Manual.*

As additional replication servers are added to a domain, the replication system administrator creates replication server interfaces (RSI), or routes, between the replication servers. Routes allow replicate transactions to “flow” from a primary replication server to a replicate replication server.

Finally, application databases are added to a domain. For each database added to the domain the following components are created:

- for primary databases, an log transfer manager (LTM), which transfers database transactions from the primary database to the replication server
- for replicate database an interface from the replicate replication server to the replicate database

Replication System Administrator (RSA)

Administering the replication system is primary the role of the Replication System Administrator (RSA). The Replication System Administrator installs, configures, and administers the replication system. Given the distributed nature of the MSS implementation this role may be performed by

different people at different locations. If this is the case, various tasks for administering Replication Server may require coordination between Replication System Administrators.

The Replication System Administrator has sa user permissions, which provides that person with the ability to execute nearly all commands in the replication system. In managing the system, the Replication System Administrator may need to coordinate with DBAs for both local and remote databases.

Replication System Administrators should be experienced Sybase DBAs and should have taken the Sybase training classes Replication System Administration and Replication Disaster Recovery Workshop. They should also have read and understood the manuals: *Replication Server Administration Guide*, *Replication Server Configuration Guide for UNIX Platforms*, *Replication Server Reference Manual*, and *Replication Trouble Shooting Guide*.

Replication System Administrator Task

The following tasks are required to maintain a replication system:

Table 4B-3. Replication System Administrator Task

Task	Roles
Installing Replication Server	Replication System Administrator (RSA)
Adding or removing a Replication Server	RSA
Starting up and shutting down Replication Server.	RSA
Configuring Replication Server	RSA
Maintaining Routes (Creating and modifying)	RSA
Managing the RSSD	RSA
Adding a primary and replicate database.	RSA
Adding login names, database users, and administering appropriate permissions	RSA
Adding replicated tables or changing table schemas. Creating and modifying replicated tables Creating and modifying replication definitions Creating and materializing subscriptions at replicate sites.	RSA
Defining data server function-string classes and function strings.	RSA
Applying database recovery procedures.	RSA
Maintaining and monitoring database connections	RSA
Monitoring Replication Server	RSA
Processing rejected transactions	RSA
Quiesing Replication Server	RSA

Task	Roles
Reconciling database inconsistencies.	RSA

DAAC DBA Replication Roles and Tasks

The Database Administrator (DBA) plays a subsidiary role by supporting some Replication Server administrator task. The following are tasks that the DBA administrators will performed at the local DAACs with respect to replication server administration.

Table 4B-4. DAAC DBA Replication Roles and Tasks

Task	Roles
Installing Replication Server	Database Administrator (DBA)
Managing the RSSD	DBA
Adding a primary and replicate database.	DBA
Adding login names, database users, and administering appropriate permissions	DBA
Adding replicated tables or changing table schemas. Creating and modifying replicated tables Creating and modifying replication definitions Creating and materializing subscriptions at replicate sites.	DBA
Defining data server function-string classes and function strings.	DBA
Applying database recovery procedures.	DBA
Processing rejected transactions	DBA
Quiesing Replication Server	DBA
Reconciling database inconsistencies.	DBA

Sybase Replication Server Installation and Setup

Sybase Replication Server 11.5.1

The Sybase Replication server software that will be delivered to the DAACs must be installed on the MSS primary machines. Other than copying the software to the directories specified in the PSR, no further action for configuring the COTS product is necessary. This installation makes the rs_subcmp utility available to the CUSTOM scripts.

CUSTOM Installation

The replication package must be installed onto the appropriate mode/machines. Instructions will be delivered with the software drop.

The following software will need to be run:

- MSS db patch
- rs_UsrInstall

MSS database patch

Since replication will only occur between tables sharing the same schema, the MSS must be run for replication to occur successfully. The database patches will add the Sybase login mss_acct_db_maint for replication to/from the SMC in addition to bringing the database into compliance with database schema requirements.

rs_UsrInstall script

The rs_UsrInstall script will need to be configured and executed at each site. Instructions will be delivered with the software drop. The DAAC installers will need to coordinate with the SMC as to setting password, servername, and database name parameters for the replication script. This script creates the script rs_UsrMain and its associated files based are the parameters entered by the installer.

The rs_UsrMain script is the script that will need to be run for replication to occur with the SMC.

Lastly, email addresses for the replication administrators (i.e. staff who need to be notified of error conditions) will need to be added to the email notification file located in the …/CUSTOM/dbms/COM/DBAdmin directory.

Other Installation

The Sybase administrator will need to update the Sybase Interfaces files on the MSS primary servers.

Daily Operations

TBD

Error Conditions

The output of the rs_subcmp utility is logged to EcMsRepSubCmp.log in the …/CUSTOM/logs directory. If an error condition is detected (by grepping the log after completion of the script) an email is sent to the addresses listed in the email notification file.

DAAC/SMC Coordination Issues

The DAACs should coordinate the following issues with the SMC and vice/versa.

- MSS Database Schema Versions
- Changes to the Sybase password for login mss_acct_db_maint

MSS Database Schema Version

When the SMC or a DAAC executes a database patch that changes the MSS User Profile table schema, the rs_UsrMain script will prevent execution of the rs_subcmp utility and this condition will be logged and email notification will be sent. Database patches to the MSS database should be coordinate through the SMC.

MSS login maintenance

If the password of the mss_acct_db_maint login is changed at the SMC, then the configuration files associated with rs_UsrMain will need to be updated at the SMC and at the DAACs to reflect the change.

If a DAAC changes the password of the user id, then that DAAC and SMC will need to update the configuration files associated with the rs_UsrMain script.

Replication Administration Software

Some of the Replication Server administration tasks will be supported by COTS and/or custom software (scripts). The COTS consists of the Sybase products Replication Server Manager (RSM) and Sybase Central, a GUI based administration tool.

Scripts will be developed for the following administration tasks in support of installing and configuring Replication Server and for installing replication server objects that are specific to the MSS application.

- Creating Routes
- Managing the RSSD
- Adding login names, database users, and permissions
- Creation of replication definitions, subscriptions, function strings and error classes
- Subscription materialization

Monitoring

The Sybase Central/RSM products will be used for the following tasks:

- Configuring Replication Server
- Modifying Routes
- Maintaining and monitoring database connections
- Monitoring Replication Server

Scripts that will be executed by the RSM will be developed to notify the RSA of the following events:

Component	*Event*
Servers	Active, Quiesed, Suspect, Hung, Shutdown, Dead, Unknown, Invalid
Routes	Change in status
Connection	Change in status
Partition	State change, size threshold exceeded
Queues	Latency threshold exceeded, size threshold exceeded
Database	Latency threshold exceeded

.

Recovery

Scripts will be developed to restore the RSSD or to bring application databases to a consistent state.

RSSD Recovery

- dumpdb
- dumptran
- logsegment threshold
- data segment threshold

MSS Database Recovery

- last chance logsegment threshold modification to disable secondary truncation point
- rs_subcmp scripts for each subscription in the domain

Sybase Central/RSM will be used for the following recovery tasks:

- Processing Rejected Transactions
- Quiesing Replication Server

Network and Security Requirements

The Sybase interface files used by the Replication Servers at each DAAC will need to be modified to locate all Sybase Replication and Data Server in the replication domain. Additionally, subscription materialization requires the same user id and password for the replicate replication server and the primary and replicate dataservers. Replication server userid and password maintenance must be coordinate across sites. Replication server supports password encryption, and this feature will

Operation Scenarios

A DAAC is added to the replication domain.

EROS DATA Center (EDC) is added to the domain, which before its installation includes, GSF, LAR, and NSC.

Task	Role	Site
The replication server software is copied to the local host.	RSA	EDC
Replication servers and MSS SQL servers are added to the interfaces files	RSA, DBA	All
The replication server executable, and its RSSD is created.	RSA, DBA	EDC
create route SMCF to EDC	RSA	EDC
The routes are verified at each site by executing the rs_helproute command on each DAAC's RSSD ASE server.	RSA	All
The rs_init utility is executed to add EDC's mss_acct_db to the operational domain. The utility connects to the ID server at SMC to obtain unique id information for the database. The rs_init utility creates an LTM start file and starts the LTM.	RSA	EDC
Create replication definition MsAcUsrProfile	RSA	EDC
Create replication definition EcAcRequest	RSA	EDC
Create, verify, and materialized subscription MsAcUsrProfile_SMC_EDC	RSA	EDC

Fault Recovery Scenarios

General Faults

In general, Sybase Replication Server is fault tolerant. Replicate database transactions start in the primary database's transaction log, are transferred from the log to the primary replication server's queue, then to the replicate replication server's queue before being applied to the replicate

databases. Database transactions are not removed from a log or a queue until the transaction has successfully moved to its next destination.

During temporary system faults, the transaction remains in its log or queue, until the fault is recovered. For example, if the replicate Sybase Server at NSC is shutdown for maintenance, replicate transactions from other DAACs are stored in the NSC's stable queue until the Sybase Server is brought back online. When NSC's replication server re-establishes its connection to the Sybase Server, the queued transactions will be applied to the replicate database in the order received. This approach is followed for all component failures.

When a failure occurs for an extended period or is of the type that causes a loss of replicate transactions (e.g. the failure of a devices supporting a queue or log), additional recovery steps must occur between sites.

EDC experiences an LTM failure.

The transaction log of the mss_acct_db at EDC is half full when the EDC's LTM suddenly, and expectedly, crashes. Meanwhile, a large number of orders are requested and the database's transaction log reaches its last-chance threshold. The threshold-stored procedure fires and forces a truncation of the transaction log. The stored procedure will log an error message in the SQL Server error log to serve notice that the truncated transactions have not been replicated. The threshold≠ stored procedure prevents the mss_acct_db database from 'freezing'; however, a recovery procedure will need to be used to forward the lost transactions to other DAACs.

The following tasks must occur:

Task	Role	Site
All client connections to the EDC SQL Server are suspended. Any transaction coming into EDC from other DAACs is queued in the EDC replication server stable device.	RSA, DBA	EDC
The EDC Replication Server's stable device is cleared of any open transactions.	RSA	EDC
The EDC mss_acct_db transaction log is dumped.	DBA	EDC
After verifying that EDC's transactions in the GSF stable queue have been processed, the rs_subcmp utility is executed to update EDC's primary fragment at GSF.	RSA	GSF
After verifying that EDC's transactions in the LAR stable queue have been processed, the rs_subcmp utility is executed to update EDC's primary fragment at LAR.	RSA	LAR
After verifying that EDC's transactions in the NSC stable queue have been processed, the rs_subcmp utility is executed to update EDC's primary fragment at NSC.	RSA	NSC
The LTM at EDC is started.	RSA	EDC

Task	Role	Site
Set the secondary truncation point at EDC to valid.	DBA	EDC
Resume client application connections to the EDC SQL Server.	DBA, RSA	EDC
Resume the DSI connection at EDC.	RSA	EDC

The GSF MSS database becomes corrupt and needs to be restored from backup.

The GSF mss_acct_db database was dumped at 12:00am. A database transaction dump executed successfully at 8:00 am. At 12:00pm, GSF's database logs become corrupt and the SQL server takes the database off-line and suspends client connections using the database.

Task	Role	Site
The LTM is shutdown.	RSA	GSF
Restart the GSF Replication Server in standalone mode.	RSA	GSF
The command admin get_generation, data_server, database is executed on the GSF Replication Server.	RSA, DBA	GSF
The command set log recovery for data_server.database is executed.	RSA	GSF
A checkpoint is issued in the mss_acct_db database.	DBA	GSF
The GSF LTM is started with the for_recovery option.	RSA	GSF
The 12:00 am database dump and the 8:00am transaction dump are loaded.	DBA	GSF
The GSF LTM is shutdown.	RSA	GSF
The command rs_zeroltm, e0mss20_srvr, mss_acct_db is executed.	RSA	GSF
The command dbcc settrunc('ltm', 'gen_id', <new_number>) is executed.	DBA	GSF
The rs_subcmp utility is executed to synchronize EDC's copy of GSF's user profile primary fragment.	RSA	EDC
The rs_subcmp utility is executed to synchronization EDC's copy of GSF's EcAcRequest primary fragment.	RSA	EDC
The rs_subcmp utility is executed to synchronize LAR's copy of GSF's user profile primary fragment.	RSA	LAR
The rs_subcmp utility is executed to synchronize NSC's copy of GSF's user profile primary fragment.	RSA	NSC
Restart the GSF Replication Server in normal mode.	RSA	GSF

Task	Role	Site
The LTM is started	RSA	GSF
Connections are resumed at the GSF SQL Server	DBA	GSF

EDC RSSD becomes corrupt and needs to be restored.

RSSD recovery is different depending on the activity that occurred since the RSSD was dumped. There are four increasingly severe levels of RSSD failure with increasingly complex recovery requirements.

Activity Since Last RSSD Dump	Procedure
No DDL activity	Basic RSSD Recovery Procedure
DDL activity, but no new routes or subscriptions created	Subscription Comparison Procedure
DDL activity, no new routes created	Subscription Re-Creation Procedure
New routes created	Deintegration/Reintegration Procedure (involves removing and reinstalling replication server)

This scenario assumes that no DDL activity occurred since the last RSSD dump. DDL commands in replication command language (RCL) include those for creating, altering, or deleting routes, replication definitions, subscriptions, function strings, functions, function-string classes, or error classes.

Tasks for Basic RSSD Recovery Procedure:

Task	Role	Site
Shutdown all RepAgents and LTM that connect to the Replication Server.	RSA	EDC
Shutdown the Replication Server if it is not down.	RSA	EDC
Restore the RSSD by loading the most recent RSSD database dump and transaction dumps.	RSA, DBA	EDC
Restart the Replication Server in standalone mode.	RSA	EDC
Log into the Replication Server and get the generation number for the RSSD.	RSA	EDC
Rebuild the Replication Server queues.	RSA	EDC

Task	Role	Site
Start all RepAgents and LTMs in recovery mode.	RSA	EDC
Check the loss messages in the Replication Server log, and in the logs of all Replication Servers with direct routes from the current Replication Server. (GSF, LAR, NSC) If a loss is detected, see the recovery procedure for scenario The GSF MSS database may have become corrupt and may need to be restored from backup.	RSA	All
Shutdown the LTM managed by the current Replication Server.	RSA	EDC
Execute the dbcc settrunc command at the Adaptive Server for the restored RSSD. Move up the secondary truncation point.	RSA, DBA	EDC
Execute the dbcc settrunc command at the Adaptive Server for the restored RSSD to set the generation number to one higher than the number returned in step 5.	RSA, DBA	EDC
Restart the Replication Server in normal mode.	RSA	EDC
Restart the RepAgents for the RSSD and the LTM in normal mode.	RSA	EDC

ASF is now part of replication domain.

Reference Document

The following are Reference documents and other information that will be help in the administration of Sybase Replication Server.

Name	Web site Address
Sybase Web Site	http://www.sybase.com/
Points of Contact web site Address	http://m0mss01.ecs.nasa.gov/smc/
Replication Server Reference Manual	http://www.sybase.com/products/datamove/
Sybase Central Installation Instruction	http://cmdm.east.hitc.com
Replication Sever Manager Installation Instruction	http://cmdm.east.hitc.com
609-CD-500-001, Release 5A Operations Tools Manual (5/99)	http://edhs1.gsfc.nasa.gov/waisdata/catalog/rel5cat.html
Database Administrators	http://www.sybase.com
Configuration Parameter Document	http://cmdm.east.hitc.com
DBA/RSA Points of Contact at web site Address	http://m0mss01.ecs.nasa.gov/smc/
313-CD-500-001, Release 5A ECS Internal Interface Control Document for the ECS Project (5/99)	http://edhs1.gsfc.nasa.gov/waisdata/catalog/rel5cat.html
25-CD-513-001, Release 5A, ECS Project Training Material Volume 13: User Services (	http://edhs1.gsfc.nasa.gov/waisdata/catalog/rel5cat.html

7/99)	